N 70-37169

NASA TECHNICAL MEMORANDUM

NASA TM X-2070

NASA TM X-2070

ORBITAL RECONNAISSANCE PROGRAM FOR VIKING CLASS MISSIONS

by Richard N. Green and Sue W. Souders

Langley Research Center
Hampton, Va. 23365

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • SEPTEMBER 1970

1. Report No. NASA TM X-2070

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
ORBITAL RECONNAISSANCE PROGRAM FOR VIKING CLASS MISSIONS

5. Report Date
September 1970

6. Performing Organization Code

7. Author(s)
Richard N. Green and Sue W. Souders

8. Performing Organization Report No.
L-7214

9. Performing Organization Name and Address
NASA Langley Research Center
Hampton, Va. 23365

10. Work Unit No.
815-10-01-20

11. Contract or Grant No.

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, D.C. 20546

13. Type of Report and Period Covered
Technical Memorandum

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A computer program has been developed to meet the needs of Project Viking in the area of orbital reconnaissance. This program is presented as an analytical tool for preliminary mission design and analysis.

The program has the capability of investigating the time history of both the spacecraft groundtrack and the camera footprint on the rotating surface of Mars. Various options are available to control where and how the photographs are taken. A sequence of photographs can be taken over a desirable lighting band, between two orbital true anomalies, or between latitudes of interest. Within the interval of desirable photography the photographs can be taken on either a time increment or a true anomaly increment. Also included is the option to take photographs on an overlap area consideration. Both vertical and nonvertical photography are available as program options. A description of the program input and output, a FORTRAN listing of the program, and samples of the input and output are included.

17. Key Words (Suggested by Author(s))
Orbit
Groundtrack
Photography
Viking

18. Distribution Statement
Unclassified – Unlimited

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of Pages
83

22. Price*
$3.00

*For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151

ORBITAL RECONNAISSANCE PROGRAM
FOR VIKING CLASS MISSIONS

By Richard N. Green and Sue W. Souders
Langley Research Center

SUMMARY

A computer program has been developed to meet the needs of Project Viking in the area of orbital reconnaissance. This program is presented as an analytical tool for preliminary mission design and analysis.

The program has the capability of investigating the time history of both the spacecraft groundtrack and the camera footprint on the rotating surface of Mars. Various options are available to control where and how the photographs are taken. A sequence of photographs can be taken over a desirable lighting band, between two orbital true anomalies, or between latitudes of interest. Within the interval of desirable photography the photographs can be taken on either a time increment or a true anomaly increment. Also included is the option to take photographs on an overlap area consideration. Both vertical and nonvertical photography are available as program options. A description of the program input and output, a FORTRAN listing of the program, and samples of the input and output are included.

INTRODUCTION

A computer program, PANDG (Photography and Groundtrack), has been developed to meet the needs of Project Viking in the area of orbital reconnaissance. Specifically, PANDG was developed in support of the Langley Viking Project Office for use in preliminary mission design and analysis.

The program has the capability of investigating the time history of both the spacecraft groundtrack and the camera footprint on the rotating surface of Mars. Various options are available to control where and how the photographs are taken. A sequence of photographs can be taken over a desirable lighting band, between two orbital true anomalies, or between latitudes of interest. Within the interval of desirable photography the photographs can be taken on either a time increment or a true anomaly increment. Also included is the option to take photographs on an overlap area consideration. Both vertical and nonvertical photography are available as program options.

The program has been designed to output either photographic data or groundtrack data or the two in combination. The groundtrack output consists of the Julian date, the calendar date, and the time elapsed from the initial true anomaly. Also included are the latitude and longitude of the subsatellite point, the altitude, inertial velocity, true anomaly, and flight-path angle of the spacecraft. Additional parameters include the lighting angle at the subsatellite point, the hour angle of the Martian vernal equinox measured from the zero meridian, the argument of periapsis, and the longitude of the ascending node of the spacecraft orbit. Equations for most of these parameters are developed herein. The photographic output consists of parameters of interest at the four corners of the camera footprint, at the midpoint of each of the four sides, and at the center point of the photograph. At each of these nine points the following parameters are output: latitude, longitude, lighting angle, slant range distance, and static resolution. One of the most important photo output parameters is the ratio of the overlap area between the present camera footprint and the previous footprint. The development of these parameters is also presented.

One advantage of PANDG is the capability to control the overlap area. Previously, some investigators have used forward overlap and side overlap as a measure of the overlap area; this definition was commonly used for Lunar Orbiter photography. However, for Mars photography, a more complete definition is needed because of the geometry created by a rapidly spinning planet. Therefore, overlap was defined as the ratio of the overlap area to the area of the previous photograph. For simplicity, the sides of the footprint were considered great circles instead of minor circles. An option is available to control the overlap between consecutive photographs.

The information necessary for the implementation of the program is contained in the appendixes of this paper. The function of the main program is outlined with a flow diagram; the purpose of each subroutine is set forth in appendix A. A FORTRAN listing is included in appendix B, in addition to sample input and output in appendix C.

SYMBOLS

$\vec{A}$ vector from center of planet to corners of previous photographic footprint on surface, kilometers

a semimajor axis, kilometers

$\vec{B}$ vector from center of planet to corners of present photographic footprint on surface, kilometers

C component of unit $\hat{C}$

$\vec{C}$	camera vector directed from spacecraft along one corner of photograph to surface, kilometers
$\hat{C}$	unit vector from spacecraft along one corner of photograph
D	component of $\vec{D}$, kilometers
$\vec{D}$	arbitrary vector within overlap area, kilometers
d	magnitude of $\vec{C}$, kilometers
E	spherical excess, degrees
E_A	spherical excess of footprint formed by $\vec{A}_1,\vec{A}_2,\vec{A}_3,\vec{A}_4$ vectors, degrees
E_C	spherical excess of overlap area, degrees
e	eccentricity
$\vec{F}$	vector from center of planet to one corner of overlap area, kilometers
f	true anomaly, degrees
H	altitude, kilometers
H_a	altitude of apoapsis, kilometers
H_p	altitude of periapsis, kilometers
h_x,h_y,h_z	components of $\hat{h}$
$\hat{h}$	unit vector in azimuth direction (heading)
$\vec{I}$	footprint intersection vector, kilometers
i	inclination, degrees
JD	Julian date, days
J_{20}	second zonal harmonic of planet
n_x,n_y,n_z	components of $\hat{n}$

$\hat{n}$	unit vector normal to orbital plane
P,Q,W	P points toward periapsis, Q is in orbital plane advanced to P by a right angle in direction of increasing true anomaly, and W completes right-handed system
r	magnitude of $\vec{r}$, kilometers
r_s	radius of surface, kilometers
r_x,r_y,r_z	components of $\hat{r}$
$\hat{r}$	unit vector in radial direction
$\vec{r}$	radius vector, kilometers
S	component of unit vector directed from planet toward Sun
$\vec{S}$	unit vector from center of planet parallel to arrival asymptote of incoming hyperbola
t	time, days
V	velocity, kilometers/second
V_∞	hyperbolic excess velocity in areocentric coordinate system, kilometers/second
V/H	horizontal velocity relative to surface divided by altitude, $second^{-1}$
X,Y,Z	Cartesian coordinate axis system
X',Y',Z'	primed Cartesian coordinate axis system
x,y,z	rectangular Cartesian coordinates, kilometers
x',y',z'	rectangular Cartesian coordinates in the primed system, kilometers (see sketch (h))
$\vec{x}$	spacecraft position vector, kilometers

α	longitude, degrees
β	angle between spacecraft position vector and camera vector, degrees (see sketch (e))
β^*	critical camera angle, degrees (see sketch (e))
γ	flight-path angle, degrees
ΔF	magnitude of $\Delta\vec{F}$
$\Delta\vec{F} = \vec{F} - \vec{D}$	
δ	latitude, degrees
η	angle between sides of overlap area, degrees (see sketch (i))
θ	hour angle, degrees
Θ	longitude of $\vec{D}$, degrees
λ	angle between radius and camera center line, degrees (see sketch (b))
μ	gravitational constant, kilometers^3/second^2
ξ	angle between sides of camera footprint, degrees (see sketch (f))
Σ	azimuth, degrees
Φ	latitude of D, degrees
ϕ	angle between radius vector and Sun vector, degrees
Ω	longitude of ascending node, degrees
ω	argument of periapsis, degrees
ψ_f	camera forward half-angle, degrees (see sketch (b))

ψ_s camera side half-angle, degrees (see sketch (b))

Subscripts:

o initial conditions

R measured in rotating coordinate system

r,h,n rectangular Cartesian components in $\hat{r},\hat{h},\hat{n}$ coordinate system

x,y,z rectangular Cartesian components in X,Y,Z coordinate system

x',y',z' rectangular Cartesian components in primed system of coordinates (see sketch (h))

Caret (^) over a symbol denotes unit vector.

Dot over a symbol denotes derivative with respect to time.

Numerical subscripts are used to distinguish similar quantities. (For example, $\vec{A}_1,\vec{A}_2,\vec{A}_3,\vec{A}_4$ are the four vectors defined as $\vec{A}$.)

METHOD OF CALCULATION

The PANDG computer program has been designed to compute areographic photographic and groundtrack data rapidly. The motion of the spacecraft about Mars has been considered Keplerian motion with the inclusion of the second zonal harmonic J_{20}, which perturbs only the longitude of the ascending node and the argument of periapsis. The program is complicated only by the variety of options which are available to the user. These options, however, are necessary to provide the flexibility needed for mission analysis. The program operates in three modes. The first mode produces only groundtrack data, the second produces only photographic data, and the third mode combines the two types of data. With the third option the groundtrack of a number of revolutions will be output with photographic output during regions of interest.

The initial state of the spacecraft can be input in one of three ways. The most obvious definition of the initial state consists of the six Keplerian orbital elements a, e, i, ω, Ω, and f. Often, however, the altitudes of periapsis and apoapsis are more convenient to input. Therefore, the option to replace a and e with H_a and H_p has been included. The third option derives the initial state from a knowledge of the incoming approach hyperbola. (See ref. 1.) To exercise this option, the right ascension and

declination of the approach asymptote $\vec{S}$ must be input along with the hyperbolic excess velocity V_∞ and the desired inclination of the spacecraft orbit. In addition, the shape of the orbit must be defined by H_a and H_p. This option assumes a deboost from a hyperbolic periapsis to an elliptical periapsis and defines the initial true anomaly of the spacecraft as zero. Since PANDG calculates data relative to the rotating surface, the initial time is important. Allowance has been made to input either an initial Julian date or an initial calendar date.

From the initial state, the spacecraft is stepped either to the point of the first groundtrack output or to the point of the first photograph, depending on which occurs first. Many times it is desirable to increment the spacecraft in time. Often, however, a true anomaly increment seems more appropriate. Therefore, both types of increments have been included as options.

If the program is exercised in one of the two photographic modes, the region over which photographs are to be taken must be specified. This region can be a Sun angle band, the section along the trajectory defined by two true anomalies, or a latitude band. Also included is the option to take a single photograph at a given latitude. Within the photographic region of interest, multiple photographs are taken. The spacing of these photographs can be specified in one of two ways, that is, either on the groundtrack step (true anomaly or time) or on an overlap area consideration. In addition to the above specifications, the photography must be defined as either vertical or nonvertical. If the nonvertical option is employed, the camera axis is rotated about the heading direction from a vertical direction through an angle λ. In other words, the camera axis is rotated out of the orbital plane by an amount of λ.

Termination of a specific case can be defined by three stop options – time, orbit number, or the central angle traversed by the spacecraft. There is no limit to the period of time which can be investigated. As the spacecraft position is incremented in time, the position of Mars is also updated with the mean orbital elements of the planet about the Sun. The location of the Martian prime meridian is also updated so that areographic parameters, expressed in the Mars centered rotating coordinate system, can be examined. Therefore, the number of days that are investigated is at the discretion of the user.

Initial State of the Spacecraft

Dependent on which input option is exercised, the six Keplerian orbital elements of the spacecraft are calculated. The first option calls for all six elements. The second option expresses the shape of the orbit in terms of H_a and H_p, the altitudes of apoapsis and periapsis, respectively. For the second option, the semimajor axis a and the eccentricity e are given by

$$a = \frac{H_a + H_p + 2r_s}{2}$$

$$e = 1 - \frac{r_s + H_p}{a}$$

where r_s is the radius of Mars. The third option derives the orbital elements from the arrival hyperbola. (See ref. 1.) The six orbital elements are then converted to inertial Cartesian components.

Groundtrack Parameters

The groundtrack output consists of a number of parameters, some of which relate to the Martian surface. To calculate these parameters, the state of the spacecraft must be expressed in the rotating equatorial coordinate system. The relationship between the two Mars centered coordinate systems, the inertial equatorial coordinate system (areocentric) and the rotating equatorial system (areographic), is through the hour angle of the Martian vernal equinox as measured from the prime meridian. The hour angle θ is given by (ref. 2)

$$\theta = \theta_o + \dot{\theta}\,\Delta t \tag{1}$$

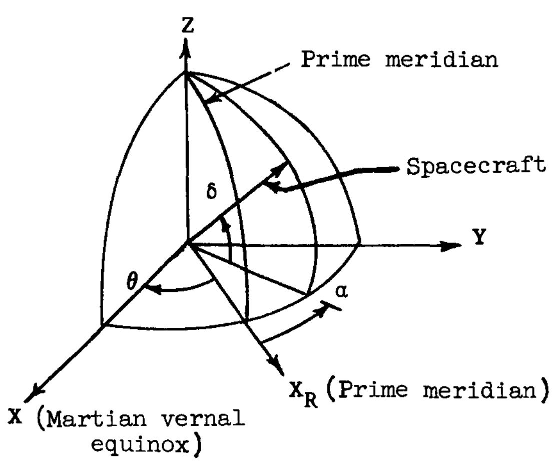
where

$$\theta_o = 149^o.475$$

$$\dot{\theta} = 350.891962 \text{ deg/day}$$

$$\Delta t = JD - 2418322.0$$

The present time is expressed in terms of Julian date JD.



Sketch (a)

It is now an easy task to express the spacecraft state in the areographic coordinate system. From sketch (a) and reference 3,

$$\left.\begin{aligned}
x_R &= x \cos\theta + y \sin\theta \\
y_R &= -x \sin\theta + y \cos\theta \\
z_R &= z \\
\dot{x}_R &= \dot{x}\cos\theta + \dot{y}\sin\theta + y_R\dot{\theta} \\
\dot{y}_R &= -\dot{x}\sin\theta + \dot{y}\cos\theta - x_R\dot{\theta} \\
\dot{z}_R &= \dot{z}
\end{aligned}\right\} \qquad (2)$$

The longitude of the subsatellite point is given by

$$\left.\begin{aligned}
\sin\alpha &= \frac{y_R}{\sqrt{x_R^2 + y_R^2}} \\
\cos\alpha &= \frac{x_R}{\sqrt{x_R^2 + y_R^2}}
\end{aligned}\right\} \qquad (3)$$

where $0^{\circ} \leqq \alpha \leqq 360^{\circ}$, and the latitude is given by

$$\delta = \sin^{-1}\left(\frac{z_R}{r}\right) \qquad (4)$$

where $-90^{\circ} \leqq \delta \leqq 90^{\circ}$ and

$$r = \sqrt{x_R^2 + y_R^2 + z_R^2}$$

The altitude is simply

$$H = r - r_s$$

and the inertial velocity of the spacecraft is given by

$$V = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}$$

To find the inertial flight-path angle γ, the vertical component of the inertial velocity vector, which is found by the dot product, is divided by the velocity, that is

$$\gamma = \sin^{-1}\left(\frac{x\dot{x} + y\dot{y} + z\dot{z}}{rV}\right)$$

where $-90^{\circ} \leqq \gamma \leqq 90^{\circ}$. The Sun angle at the subsatellite point is defined as the angle between the spacecraft radius vector and the vector to the Sun. When the Sun is directly overhead, the Sun angle is zero. To obtain the unit Sun vector, a series of subroutines

which utilize the mean orbital elements of Mars about the Sun is exercised. A discussion of this procedure is given in reference 1. Once the Sun vector has been obtained, the vector dot product yields the Sun angle at the subsatellite point, that is

$$\phi = \cos^{-1}\left(\frac{xS_x + yS_y + zS_z}{r}\right) \tag{5}$$

where $0^{\circ} \leqq \phi \leqq 180^{\circ}$ and (S_x,S_y,S_z) is the unit vector directed toward the Sun. Also included in the set of groundtrack parameters is V/H, or the horizontal velocity relative to the surface divided by the altitude of the spacecraft. The horizontal velocity can be found by considering the magnitude of the vector cross product of the relative velocity vector and the radius vector, that is

$$V(\text{horizontal}) = \frac{\left|\vec{x}_R \times \dot{\vec{x}}_R\right|}{r} = V_R \sin(90^{\circ} - \gamma)$$

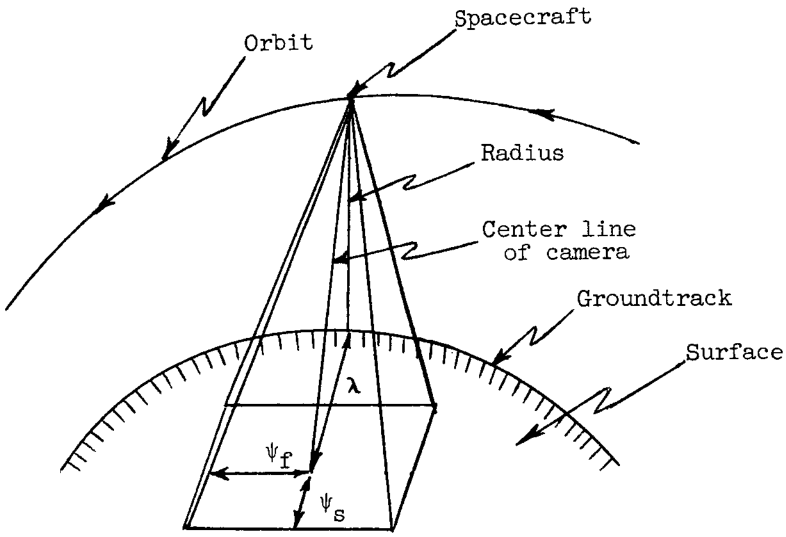
where $(90^{\circ} - \gamma)$ is the angle between the two vectors, or the complement of the relative flight-path angle. Therefore, V/H is given by

$$V/H = \frac{\left[\left(y_R\dot{z}_R - \dot{y}_R z_R\right)^2 + \left(\dot{x}_R z_R - x_R\dot{z}_R\right)^2 + \left(x_R\dot{y}_R - \dot{x}_R y_R\right)^2\right]^{1/2}}{rH}$$

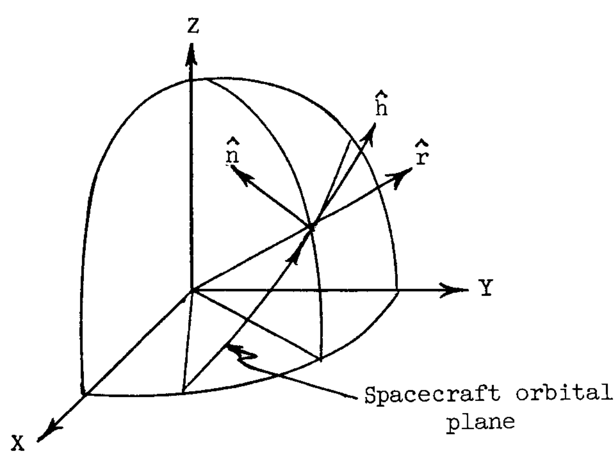
In addition to the above groundtrack parameters are the following: Julian date, calendar date, orbit time, orbit number, and the true anomaly of the spacecraft.

Photographic Parameters

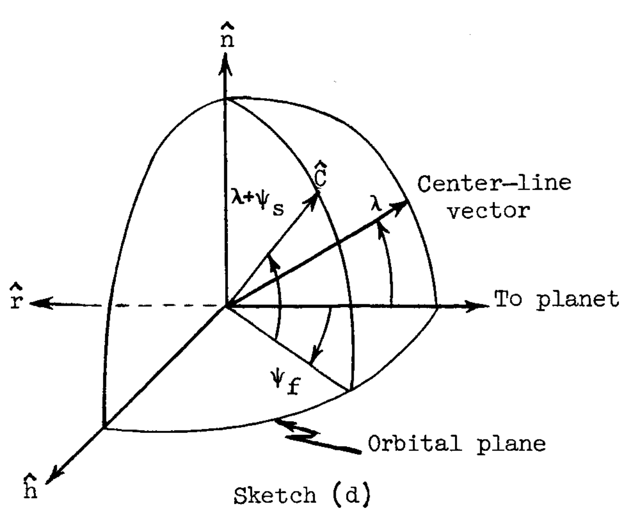
The main objects of interest at a photographic point are the camera footprint on the surface (that area of the surface photographed by the onboard camera) and the set of parameters which are evaluated at each of the four corners of the footprint and at the midpoint of each side. The location of the footprint, however, depends on the camera system and its characteristics. The camera system discussed herein is defined as being alined with the inertial orbital plane and is described by a forward half-angle ψ_f and a side half-angle ψ_s. (See sketch (b).) In addition, the capability to rotate the camera center line out of the plane of motion by an angular amount λ has also been included. For vertical photography, the camera center line is directed downward along the radius vector ($\lambda = 0$). To facilitate the calculation of the points where the four corner vectors pierce the surface, consider the spacecraft-centered coordinate system $\hat{r},\hat{h},\hat{n}$ (sketch (c)). The unit vector $\hat{r}$ is directed along the radius vector to the spacecraft, and the vector $\hat{h}$ (heading) is in the azimuth direction. The unit vector $\hat{n}$ is normal to the plane of motion, that is, $\hat{n} = \hat{r} \times \hat{h}$. The unit vector $\hat{C}$ directed from the spacecraft along one of the corners of the photograph, defined by a positive forward half-angle ψ_f and a positive side half-angle ψ_s, is illustrated in sketch (d). Also, a positive out-of-plane angle λ has



Sketch (b)



Sketch (c)



Sketch (d)

been considered. The components of the corner vector in the $\hat{r},\hat{h},\hat{n}$ system are therefore given by

$$\left.\begin{aligned} C_r &= -\cos\left(\lambda + \psi_s\right)\cos\psi_f \\ C_h &= \cos\left(\lambda + \psi_s\right)\sin\psi_f \\ C_n &= \sin\left(\lambda + \psi_s\right) \end{aligned}\right\} \tag{6}$$

The corner vector in the X,Y,Z system remains to be expressed. The unit vector $\hat{r}$ expressed in the X,Y,Z system is

$$\hat{r} = \frac{\vec{x}}{r} = \left(r_x, r_y, r_z\right)$$

and the unit vector $\hat{n}$ is given by

$$\hat{n} = \frac{\hat{r} \times \dot{\vec{x}}}{\left|\hat{r} \times \dot{\vec{x}}\right|} = \left(n_x, n_y, n_z\right)$$

The unit vector $\hat{h}$ is

$$\hat{h} = \hat{n} \times \hat{r} = \left(h_x, h_y, h_z\right)$$

Therefore, the unit vector $\hat{C}$ in the X,Y,Z system is

$$\begin{bmatrix} C_x \\ C_y \\ C_z \end{bmatrix} = \begin{bmatrix} r_x & h_x & n_x \\ r_y & h_y & n_y \\ r_z & h_z & n_z \end{bmatrix} \begin{bmatrix} C_r \\ C_h \\ C_n \end{bmatrix}$$

From sketch (e), the angle β between the radius vector $\vec{x}$ and the corner vector $\hat{C}$ is given by

$$\beta = \cos^{-1}\left(-\hat{x} \cdot \hat{C}\right)$$

where $0^\circ \leqq \beta \leqq 180^\circ$. If β exceeds the value of β^*, the corner vector does not pierce the surface. From sketch (e),

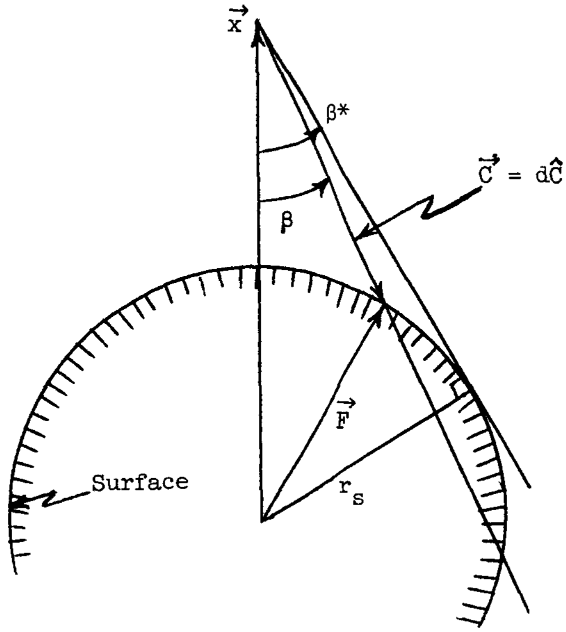
$$\beta^* = \sin^{-1}\left(\frac{r_s}{r}\right)$$

where $0^\circ \leqq \beta^* \leqq 180^\circ$. Hence, for the corner vector to pierce the surface, $\beta < \beta^*$. The slant range distance d is given by the law of cosines as

$$r_s^2 = r^2 + d^2 - 2rd\cos\beta$$

or

$$d = \frac{2r\cos\beta \pm \sqrt{4r^2\cos^2\beta - 4\left(r^2 - r_s^2\right)}}{2}$$



Sketch (e)

Since the extended corner vector pierces the surface at two points and since the smaller of the two solutions for d is the desired solution, the radical is negative and

$$d = r \cos \beta - \sqrt{r^2 \cos^2 \beta - r^2 + r_s^2}$$

Therefore, the point at which the camera corner vector pierces the surface of the planet is given by

$$\vec{F} = \vec{x} + \vec{C}$$

where

$$\vec{C} = d\hat{C}$$

The other three corners are found in a similar manner by considering $\pm\psi_s$ and $\pm\psi_f$ in equation (6). The latitude and longitude of the corner vector $\vec{F}$ are found with equations (1) to (4) and the Sun angle with equation (5).

Camera Footprint Overlap

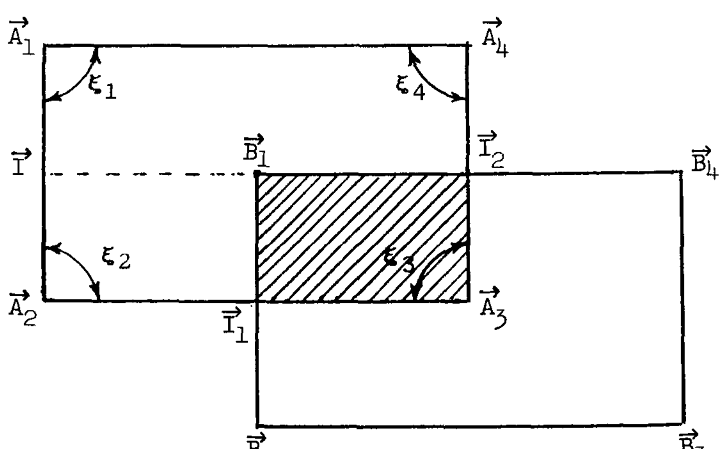
Another photographic parameter of interest is the camera footprint overlap. The camera overlap area, as defined in PANDG, is the surface area contained in both the present camera footprint and the previous camera footprint. The ratio of this overlap area to the area of the previous footprint is designated as the camera footprint overlap or simply as the overlap. This parameter is important in designing a sequence of photographs so that they can be overlaid to form a map of the area photographed. In addition, overlap is necessary to obtain stereo effects from the photographs. Previously, some investigators have used forward overlap and side overlap as a measure of the overlap area. This

definition was commonly used for Lunar Orbiter photography. However, for Mars photography, a more complete definition is needed because of the geometry created by the higher spin rate of the planet. In certain instances the surface with the camera footprint rotates faster than the spacecraft revolves in its orbit. Therefore, the footprint is ahead of the spacecraft, and forward overlap would not be adequate.

The camera footprint is that area of the surface photographed by the onboard camera. As shown previously, the four corner points or vectors can be found. The sides of the footprint are actually arcs of minor circles, which are difficult to analyze mathematically. To simplify computations, the sides have been considered as great circles. This approximation is justifiable for photography taken at low altitudes (small footprint). However, as the footprint becomes large, the error associated with this approximation becomes more pronounced. Since most of the photography will be at low altitudes, this simplification seems reasonable.

The overlap is defined by eight corner vectors. (See sketch (f).) The four vectors $\vec{A}_1$, $\vec{A}_2$, $\vec{A}_3$, and $\vec{A}_4$ represent the four corners of the previous footprint. Similarly, the present footprint is defined by the four B vectors. For the particular geometry shown, the overlap area has four corners which are typed as either interior vectors or intersection vectors. Corner vectors of the overlap area which are also corner vectors of one of the footprints are classed as interior vectors. Examples of these are $\vec{B}_1$ and $\vec{A}_3$. An intersection vector occurs at the corner formed by the intersection of one side from each of the two footprints. Examples of these are $\vec{I}_1$ and $\vec{I}_2$.

The surface area of the previous footprint is found by considering the number of spherical degrees in the footprint. Since the entire spherical surface of Mars contains 720 spherical degrees and since the surface area is easily obtainable, the area of the



Sketch (f)

footprint follows from the number of spherical degrees it contains. However, to obtain the overlap, which is a ratio of areas, only the computation of the spherical degrees contained in the area is necessary and not the area itself. The number of spherical degrees in an area is equal to its spherical excess. By definition, the spherical excess of a spherical polygon is the difference between the sum of its angles and the sum of the angles of a plane polygon having the same number of sides. (See ref. 4.) Therefore, the spherical excess E_A of the footprint formed by the A vectors is given by

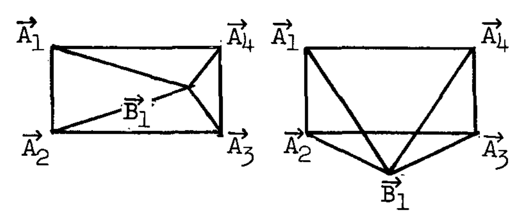
$$E_A = \left(\xi_1 + \xi_2 + \xi_3 + \xi_4\right) - 360^{\circ}$$

The ξ angles are the angles between the great circular arcs which connect the corner points. These angles are most easily found by considering the vector normal to each side and computing the angles between the normals. Therefore, ξ_1 is given by

$$\xi_1 = \cos^{-1}\left[\frac{\left(\vec{A}_1 \times \vec{A}_4\right) \cdot \left(\vec{A}_1 \times \vec{A}_2\right)}{\left|\vec{A}_1 \times \vec{A}_4\right|\left|\vec{A}_1 \times \vec{A}_2\right|}\right]$$

where $0^{\circ} \leqq \xi_1 \leqq 180^{\circ}$. The other three ξ angles are found in a similar manner.

Prior to the calculation of the spherical excess of the overlap area, the corner vectors must be found. First, the B vectors are examined to determine if they are interior vectors. If $\vec{B}_1$ is an interior vector, the sum of the spherical excesses of the four triangles $\vec{B}_1$-$\vec{A}_1$-$\vec{A}_2$, $\vec{B}_1$-$\vec{A}_2$-$\vec{A}_3$, $\vec{B}_1$-$\vec{A}_3$-$\vec{A}_4$, $\vec{B}_1$-$\vec{A}_4$-$\vec{A}_1$ will be equal to the spherical excess of the footprint formed by A vectors. However, if these excesses are not equal, the vector considered is not an interior vector but lies outside of the footprint area. (See sketch (g).) The spherical excess of the triangle $\vec{B}_1$-$\vec{A}_1$-$\vec{A}_2$ is given by



Sketch (g)

$$E = \cos^{-1}\left[\frac{\left(\vec{A}_1 \times \vec{B}_1\right) \cdot \left(\vec{A}_1 \times \vec{A}_2\right)}{\left|\vec{A}_1 \times \vec{B}_1\right|\left|\vec{A}_1 \times \vec{A}_2\right|}\right] + \cos^{-1}\left[\frac{\left(\vec{A}_2 \times \vec{A}_1\right) \cdot \left(\vec{A}_2 \times \vec{B}_1\right)}{\left|\vec{A}_2 \times \vec{A}_1\right|\left|\vec{A}_2 \times \vec{B}_1\right|}\right] + \cos^{-1}\left[\frac{\left(\vec{B}_1 \times \vec{A}_2\right) \cdot \left(\vec{B}_1 \times \vec{A}_1\right)}{\left|\vec{B}_1 \times \vec{A}_2\right|\left|\vec{B}_1 \times \vec{A}_1\right|}\right] - 180^{\circ}$$

The spherical excesses of the other three spherical triangles are found in a similar manner, and the sum indicates whether $\vec{B}_1$ is an interior point. This sequence is then repeated for both the A and the B vectors until all interior vectors are found.

Once the interior vectors are found, the geometry is examined to determine if any intersection vectors exist. First, the side joining $\vec{A}_1$ and $\vec{A}_2$ is examined to determine if any of the sides formed by B vectors intersect it. The intersection vector is found by the vector cross product of the normals, that is

$$\vec{I} = \pm\left(\vec{A}_1 \times \vec{A}_2\right) \times \left(\vec{B}_1 \times \vec{B}_4\right)$$

where $\vec{I}$ is the intersection vector and is multivalued since great circles intersect at two points. For $\vec{I}$ to be a corner vector of the overlap area, it must lie between $\vec{A}_1$ and $\vec{A}_2$ and also between $\vec{B}_1$ and $\vec{B}_4$. If the sum of the angles between $\vec{A}_1$ and $\vec{I}$ and between $\vec{I}$ and $\vec{A}_2$ equals the angle between $\vec{A}_1$ and $\vec{A}_2$, then $\vec{I}$ is surely between $\vec{A}_1$ and $\vec{A}_2$. The same must be true for the B vectors. Therefore, for $\vec{I}$ to be an intersection vector,

$$\cos^{-1}\left[\frac{\vec{A}_1 \cdot \vec{I}}{\left|\vec{A}_1\right|\left|\vec{I}\right|}\right] + \cos^{-1}\left[\frac{\vec{I} \cdot \vec{A}_2}{\left|\vec{I}\right|\left|\vec{A}_2\right|}\right] = \cos^{-1}\left[\frac{\vec{A}_1 \cdot \vec{A}_2}{\left|\vec{A}_1\right|\left|\vec{A}_2\right|}\right]$$

and

$$\cos^{-1}\left[\frac{\vec{B}_1 \cdot \vec{I}}{\left|\vec{B}_1\right|\left|\vec{I}\right|}\right] + \cos^{-1}\left[\frac{\vec{I} \cdot \vec{B}_4}{\left|\vec{I}\right|\left|\vec{B}_4\right|}\right] = \cos^{-1}\left[\frac{\vec{B}_1 \cdot \vec{B}_4}{\left|\vec{B}_1\right|\left|\vec{B}_4\right|}\right]$$

Both values of $\vec{I}$ must be examined as candidates for an intersection vector. As can be seen in sketch (f), $\vec{I}$ satisfies the first of the two criteria but fails the second. Therefore, $\vec{I}$ is not a corner vector of the overlap area. In turn, each of the four sides of the footprint formed by B vectors is examined in combination with the four sides of the footprint formed by A vectors. For the geometry of sketch (f), the intersection vectors $\vec{I}_1$ and $\vec{I}_2$ are found in this manner.

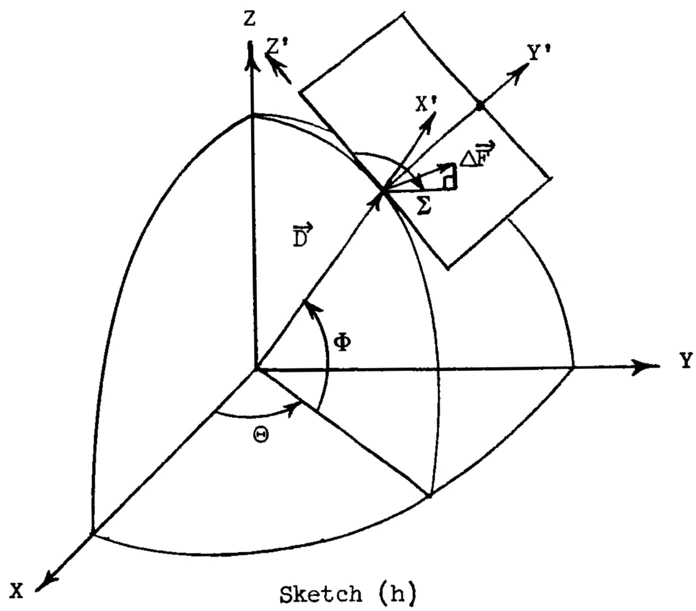
Once all the corner vectors of the overlap area are found, the overlap area is determined. However, the interior and intersection vectors found are not necessarily in clockwise or counterclockwise order. To facilitate the calculation of the overlap area, the corner vectors are ordered by considering the azimuth angle of each vector referenced to an arbitrary vector within the overlap area. The corner vectors are redefined as $\vec{F}_i$ where $i = 1, 2, 3, \ldots, n$, and the reference vector $\vec{D}$ is formed by averaging the first two corner vectors, that is

$$\vec{D} = \frac{\vec{F}_1 + \vec{F}_2}{2}$$

The vectors from $\vec{D}$ to each of the corner vectors are given by

$$\Delta\vec{F}_i = \vec{F}_i - \vec{D}$$

where $i = 1, 2, 3, \ldots, n$.



Sketch (h)

The azimuth angle is found by considering the geometry of sketch (h). The primed axis system is alined such that the X'-axis is in the radial direction, the Z'-axis is perpendicular to X' and in a northerly direction, and the Y'-axis completes the triad. The rotation from the unprimed system to the primed system is accomplished by

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos\Phi & 0 & \sin\Phi \\ 0 & 1 & 0 \\ -\sin\Phi & 0 & \cos\Phi \end{bmatrix} \begin{bmatrix} \cos\Theta & \sin\Theta & 0 \\ -\sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

where

$$\sin\Theta = \frac{D_y}{\sqrt{D_x^2 + D_y^2}}$$

$$\cos\Theta = \frac{D_x}{\sqrt{D_x^2 + D_y^2}}$$

$$\sin\Phi = \frac{D_z}{\sqrt{D_x^2 + D_y^2 + D_z^2}}$$

$$\cos\Phi = \sqrt{1 - \sin^2\Phi}$$

Therefore, $\Delta\vec{F}$ can be expressed in the primed system as

$$\Delta F_{x'} = \Delta F_x \cos\Phi \cos\Theta + \Delta F_y \cos\Phi \sin\Theta + \Delta F_z \sin\Phi$$

$$\Delta F_{y'} = -\Delta F_x \sin\Theta + \Delta F_y \cos\Theta$$

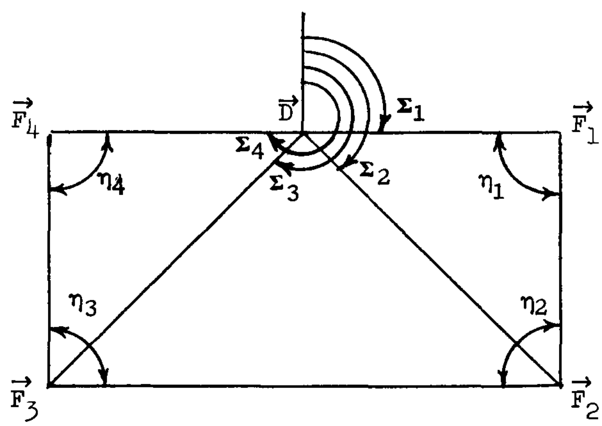
$$\Delta F_{z'} = -\Delta F_x \sin\Phi \cos\Theta - \Delta F_y \sin\Phi \sin\Theta + \Delta F_z \cos\Phi$$

and the azimuth angle Σ is given by

$$\sin\Sigma = \frac{\Delta F_{y'}}{\sqrt{\Delta F_{y'}^{\,2} + \Delta F_{z'}^{\,2}}}$$

$$\cos\Sigma = \frac{\Delta F_{z'}}{\sqrt{\Delta F_{y'}^{\,2} + \Delta F_{z'}^{\,2}}}$$

where $0^\circ \leqq \Sigma \leqq 360^\circ$. In this manner the azimuth angle for each of the corner vectors $\Delta\vec{F}_i$ is found and ordered such that they are clockwise (sketch (i)).



Sketch (i)

The spherical excess E_C of the overlap area is now given as

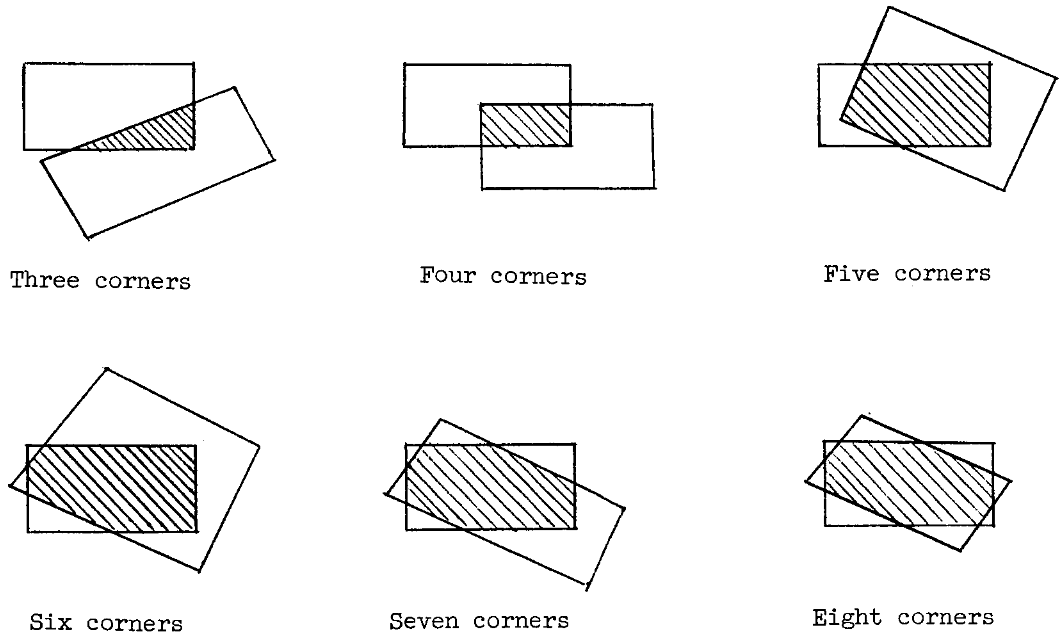
$$E_C = (\eta_1 + \eta_2 + \eta_3 + \eta_4) - 360^\circ$$

where the η angles are found in the same manner as the ξ angles. The overlap, or the ratio of the overlap area to the area of the previous footprint, is simply

$$\text{Overlap} = \frac{E_C}{E_A}$$

where the range of the overlap is from 0 to 1.

The number of corner vectors comprising the overlap area is not necessarily four but can vary from three to eight. Examples of various geometries are shown in sketch (j). Although the method described was derived by considering a four-cornered overlap area, which is the most frequent case, the procedure is applicable for all geometries. The computer program, PANDG, has the capability to calculate the overlap for all six possible geometries.



Sketch (j)

RESULTS AND DISCUSSION

The equations developed herein and the program options outlined have been incorporated into the computer program designated PANDG. The program was written in FORTRAN IV language for a digital computer and contains a main program and 31 subroutines. The information necessary for the implementation of the program is contained in the appendixes of this paper. Figure 1 is an example of the type of plots which can result from the execution of PANDG.

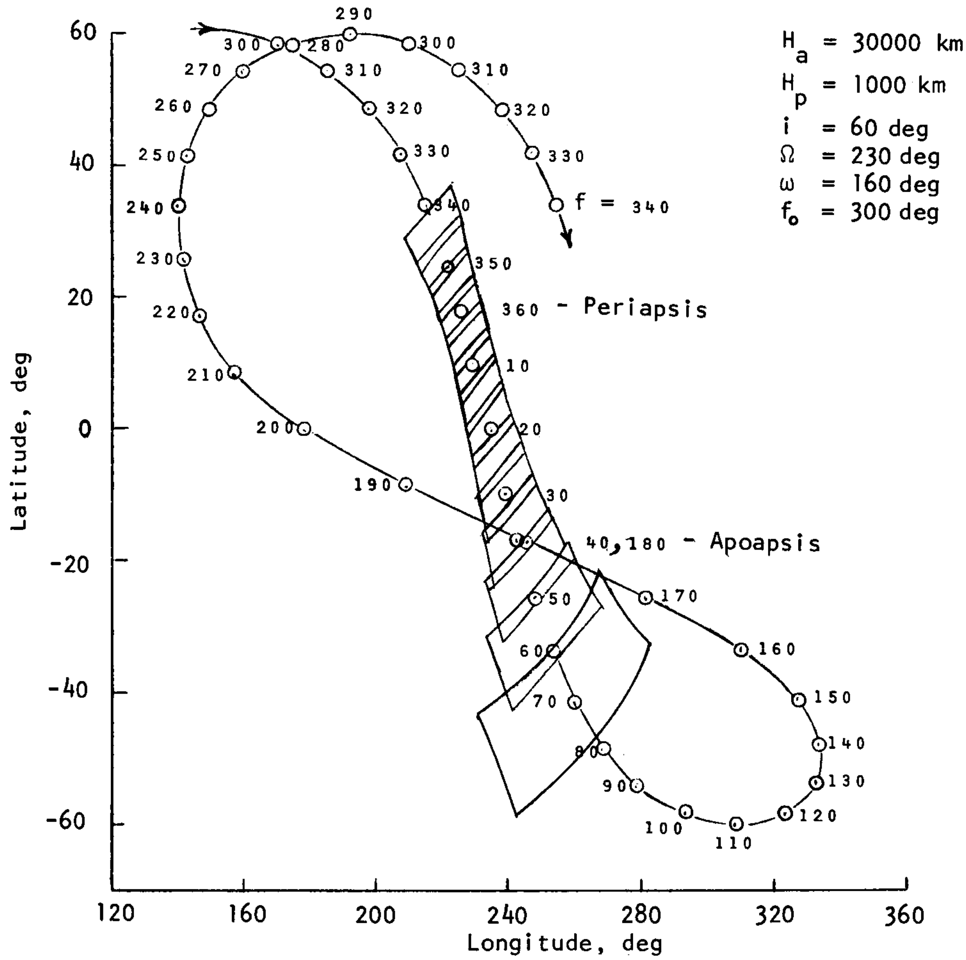


Figure 1.- Photography footprints between latitudes of -25° and 30° and groundtrack for a satellite orbit about Mars.

The groundtrack of the spacecraft has been traced by the subsatellite points at 10° intervals of true anomaly. Also shown is a series of camera footprints which resulted from the choice of several program options. The region of photographic coverage was designated as being between the latitudes of -25° and 30°. Within this region, vertical photography was taken on an overlap consideration. The first in the sequence of footprints occurred at the point where the camera center line crossed the 30° latitude line. Photographs were then taken such that a 25-percent overlap existed between consecutive frames. Photographic coverage continued until all four corners of the footprint were out of the latitude band. PANDG then continued to output groundtrack data until one of the program stop conditions was reached. For the particular case considered herein, the program stop occurred when 400° of central angle had been traversed by the spacecraft.

CONCLUDING REMARKS

A computer program has been developed to meet the needs of Project Viking in the area of orbital reconnaissance. The program is intended as an analytical tool for preliminary mission design and analysis.

Langley Research Center,
National Aeronautics and Space Administration,
Hampton, Va., June 2, 1970.

APPENDIX A

PROGRAM DESCRIPTION

The program PANDG has been written entirely in FORTRAN IV computer language for the Control Data 6600 digital computer and contains a main program and 31 subroutines. The modular form in which the program was written resulted in a field length (storage requirement) of 60 000_8. A brief statement of the purpose of each subroutine contained in the program follows.

CALJUL	Converts calendar date to Julian date
CARCON	Converts Cartesian coordinates to conic elements
CARSPH	Converts Cartesian coordinates to spherical coordinates
CONCAR	Converts conic elements to Cartesian coordinates
DOT	Calculates the angle between two vectors
EEARTH	Calculates the mean heliocentric position and velocity of Earth
EMARS	Calculates the mean heliocentric position and velocity of Mars
EULER	Performs an Euler rotation
GPRINT	Calculates and writes groundtrack data
JULCAL	Converts Julian date to calendar date
LATLNG	Converts Cartesian position to latitude and longitude
LOGIC	Examines program options to determine whether the next event is a groundtrack or photography event and calculates the time increment and true anomaly increment to the next event
OPRINT	Writes program inputs
ORBIT	Calculates Keplerian orbital elements for a periapsis-to-periapsis deboost

APPENDIX A

OVERLAP	Calculates the time increment to the next camera footprint for a given overlap ratio
PPRINT	Calculates and writes photography data
PRATIO	Calculates the overlap ratio between two footprints
PRECES	Transforms mean Earth equinox and equator coordinates from one epoch to another epoch
QADRAT	Solves an equation of the form $AX^2 + BX + C = 0$ for the real roots
RAY	Calculates the point at which the camera vector pierces the surface of the planet
RECEQ	Rotates a vector from the coordinate system of the mean equinox and ecliptic of date to the coordinate system of the mean Earth equinox and equator of date
REQMEQ	Rotates a vector from the coordinate system of the mean Earth equinox and equator of date to the coordinate system of the mean Mars equinox and equator of date
REQPEQ	Rotates a vector from the coordinate system of the mean Earth equinox and equator of date to the coordinate system of the mean planet equinox and equator of date
RXYZPQW	Rotates a vector from the X,Y,Z to the PQW coordinate system
STDCASE	Defines program constants and standard input data
SUNBAND	Calculates the two positions in orbit which correspond to a given lighting angle
TCONIC	Calculates the time from periapsis passage for a given true anomaly
TINVS	Converts mean anomaly to eccentric and true anomaly
UPDATE	Updates all time dependent parameters such as true anomaly, hour angle, and so forth

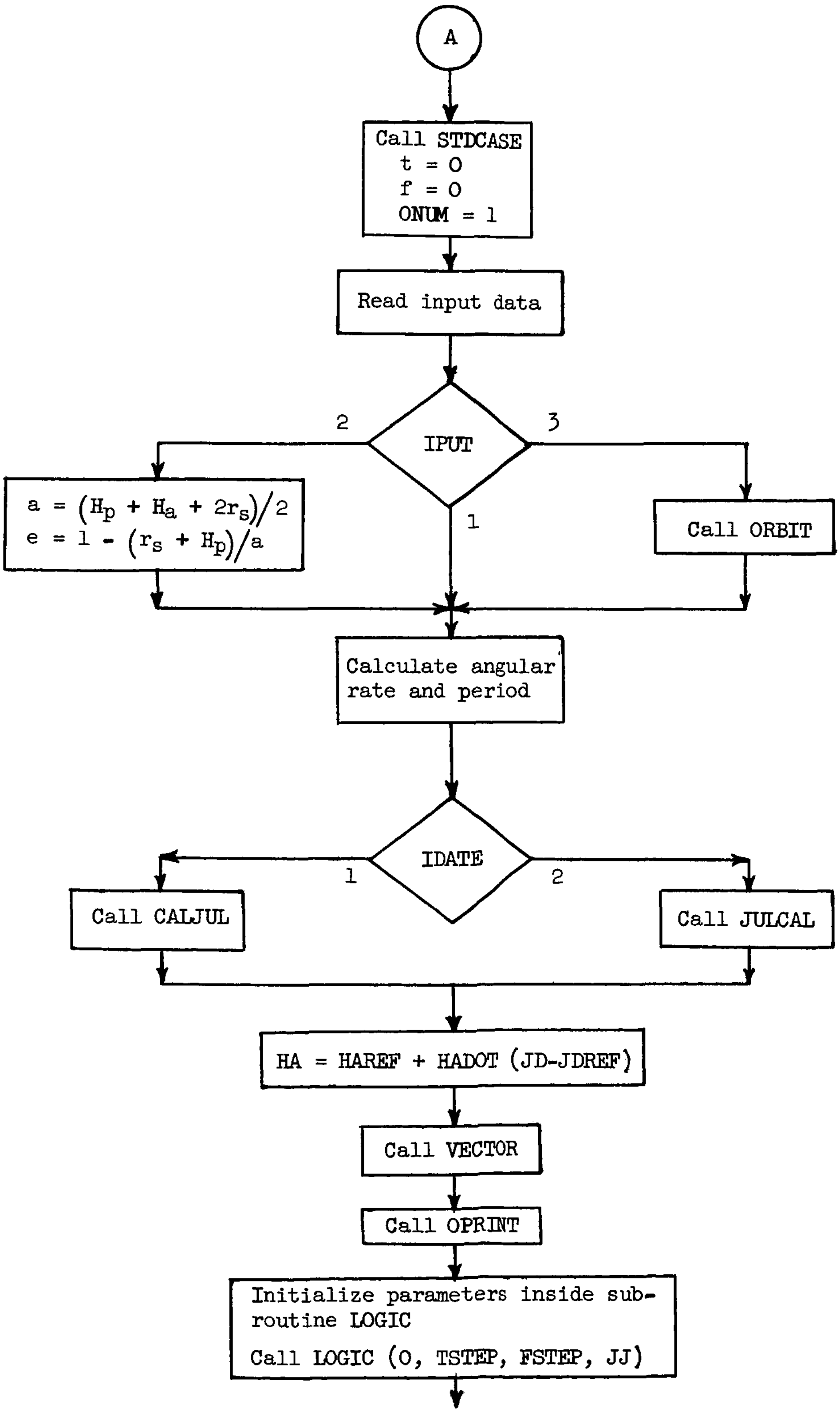
VECTOR Calculates the positions of the Sun, Earth, and Canopus in the coordinate system of the center of the planet and the planet equator

WELLS Updates the argument of periapsis and the longitude of the ascending node

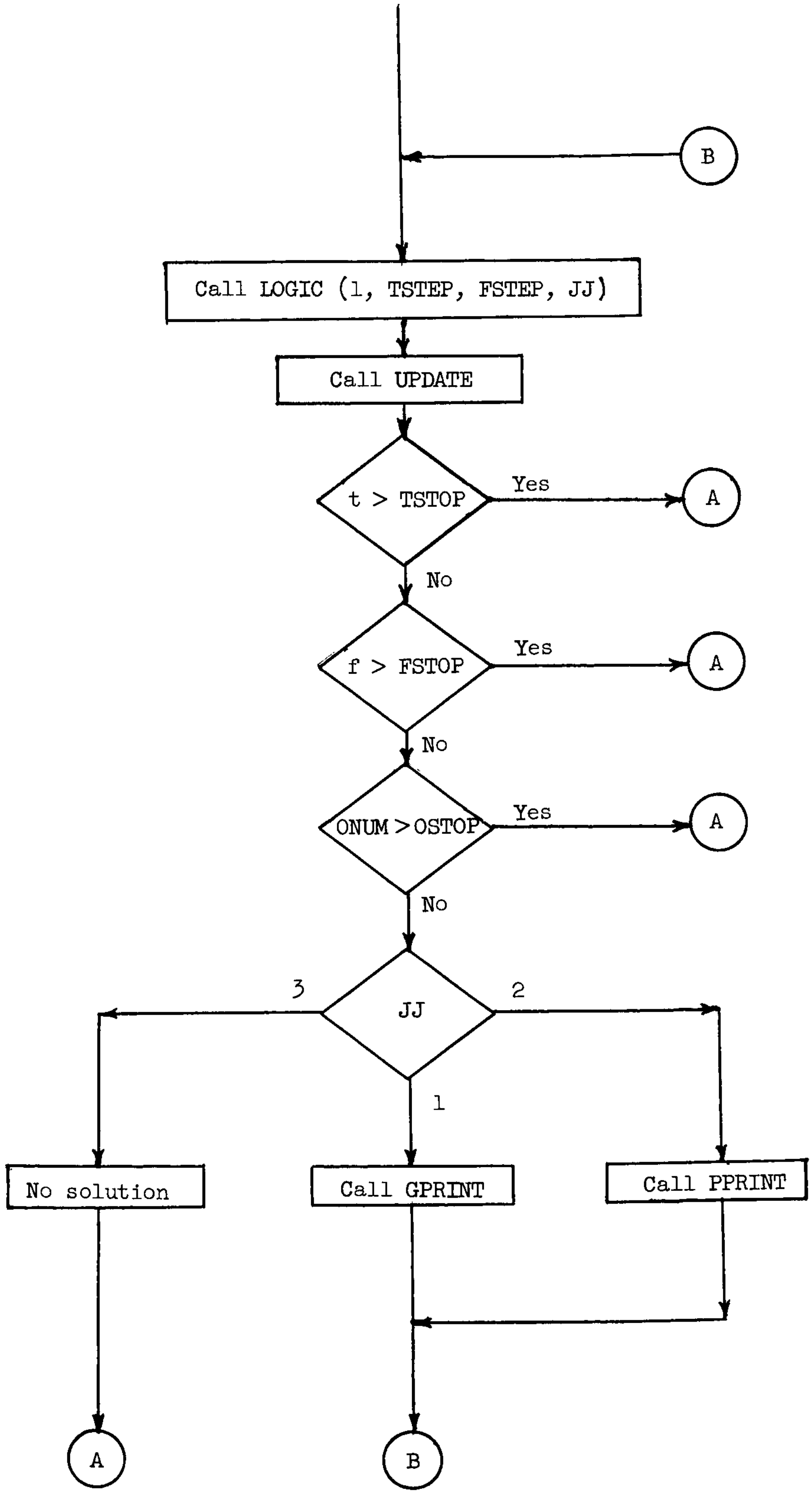
The main program is relatively simple, since its main purpose is administration. The subroutine STDCASE is called first to define the program constants, to initialize various parameters, and to define any standard data which it might contain. Afterwards, the input data are read into the program by means of a FORTRAN IV namelist. Only two options are considered in the main program. The input option (IPUT = 1, 2, 3) denotes the set of parameters input to describe the initial state of the spacecraft. The other option (IDATE = 1, 2) denotes which date was used; the initial date can be input as either a calendar date or a Julian date. Dependent on the options, the initial orbital elements and the initial Julian date are calculated. In addition, the initial hour angle HA and the unit Sun vector (subroutine VECTOR) are calculated. Afterwards, subroutine OPRINT is called to write all the program inputs.

At this point the program is ready to perform its primary task of calculating photographic and groundtrack data. Subroutine LOGIC is called to determine whether the next event is a groundtrack event or a photographic event. This information is contained in the output parameter JJ. The time increment TSTEP and the true anomaly increment FSTEP to the next event are also determined inside LOGIC. Next, subroutine UPDATE is exercised to update all time-dependent parameters according to TSTEP and FSTEP. Before processing the next event, however, the program stop conditions are examined. If the time has exceeded the time stop or the central angle has exceeded its limit or the orbit-number stop has been reached, then the case is terminated and a new set of input data are read. If none of these conditions are met, then either the groundtrack data or the photographic data are calculated and output. The program then proceeds to determine the next event. This cycle is repeated until the case is complete. A flow diagram of the main program follows.

APPENDIX A


A
Call STDCASE
t = 0
f = 0
ONUM = 1
Read input data
IPUT
2
3
1
$a = (H_p + H_a + 2r_s)/2$
$e = 1 - (r_s + H_p)/a$
Call ORBIT
Calculate angular rate and period
IDATE
1
2
Call CALJUL
Call JULCAL
HA = HAREF + HADOT (JD-JDREF)
Call VECTOR
Call OPRINT
Initialize parameters inside subroutine LOGIC
Call LOGIC (0, TSTEP, FSTEP, JJ)

APPENDIX A

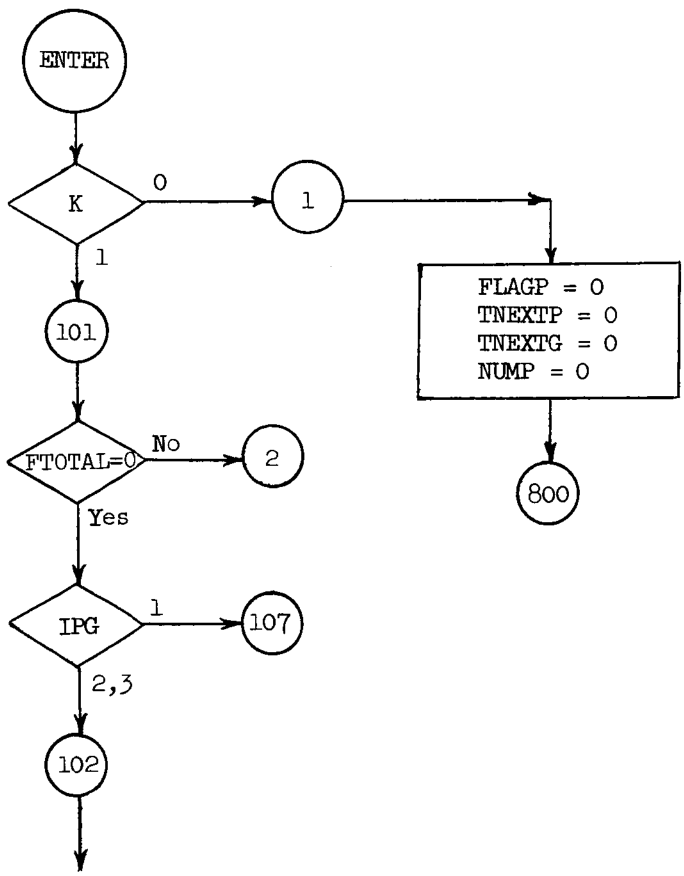


B
Call LOGIC (1, TSTEP, FSTEP, JJ)
Call UPDATE
t > TSTOP
Yes
A
No
f > FSTOP
Yes
A
No
ONUM > OSTOP
Yes
A
No
JJ
3
2
1
No solution
Call GPRINT
Call PPRINT
A
B

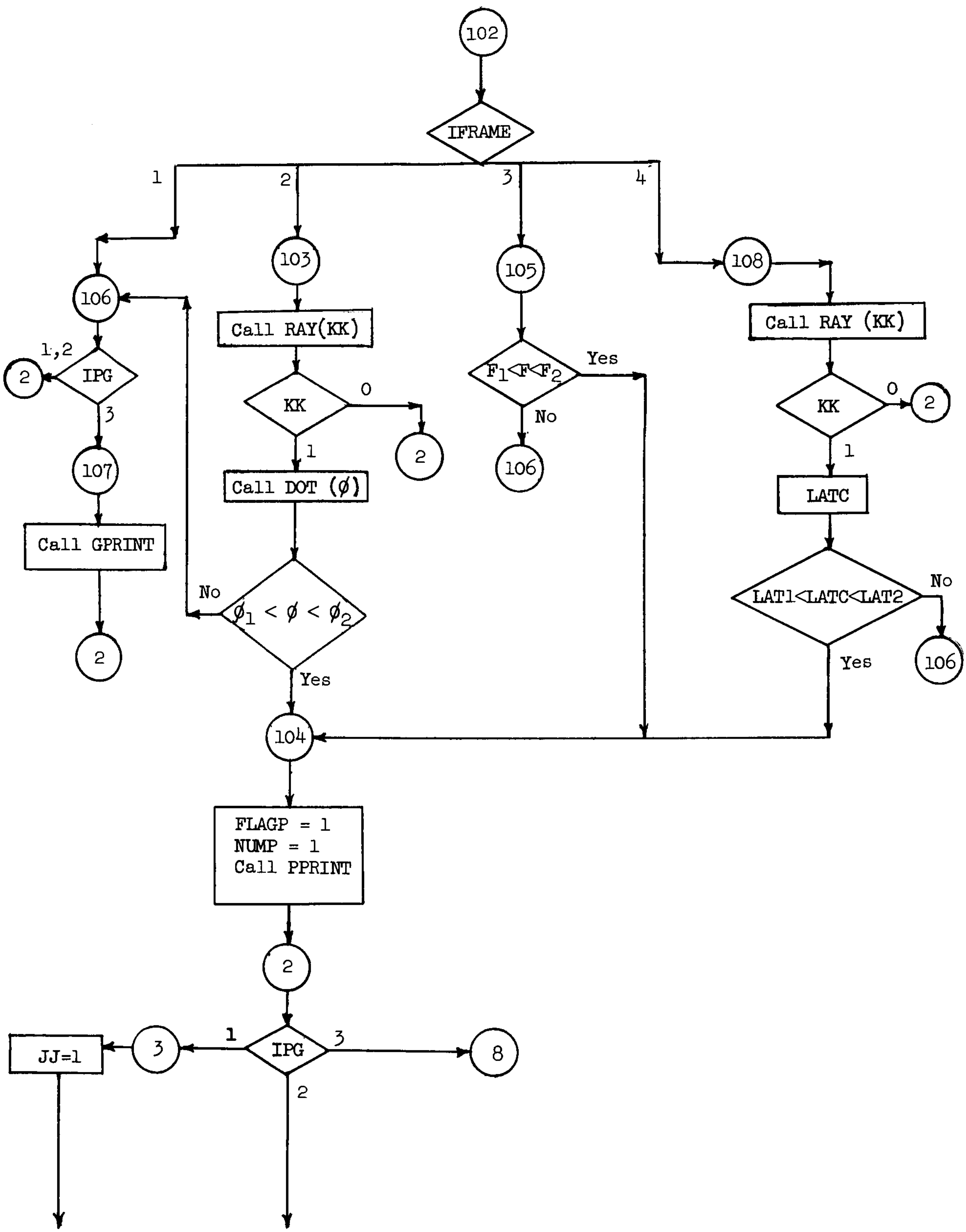
APPENDIX A

The program logic is contained in the subroutine LOGIC. The purpose of the subroutine is to examine the program options in order to determine whether the next event is a photographic event or a groundtrack event. In addition, it calculates the time and true anomaly increments to the next event. The logic involved in this calculation is rather complicated, as can be seen from the flow diagram of the subroutine. A number of different options are examined in LOGIC. The option IPG determines whether the program generates groundtrack data (IPG = 1), photographic data (IPG = 2), or the two in combination (IPG = 3). If groundtrack data are generated, then ISTEP determines whether this data is output on time (ISTEP = 1) or true anomaly (ISTEP = 2). If photographic data are generated, the camera system is defined as either vertical (IPHOTO = 0) or nonvertical (IPHOTO = 1). For a single photograph, ISINGLE = 1. Usually, however, a series of photographs is taken. The region of photographic interest is defined by the IFRAME option. Within this region the photographs can be taken on either an overlap consideration (IOLAP = 1) or a constant step (IOLAP = 0). Subroutine LOGIC considers all these different options to determine the next event. The flow diagram follows.

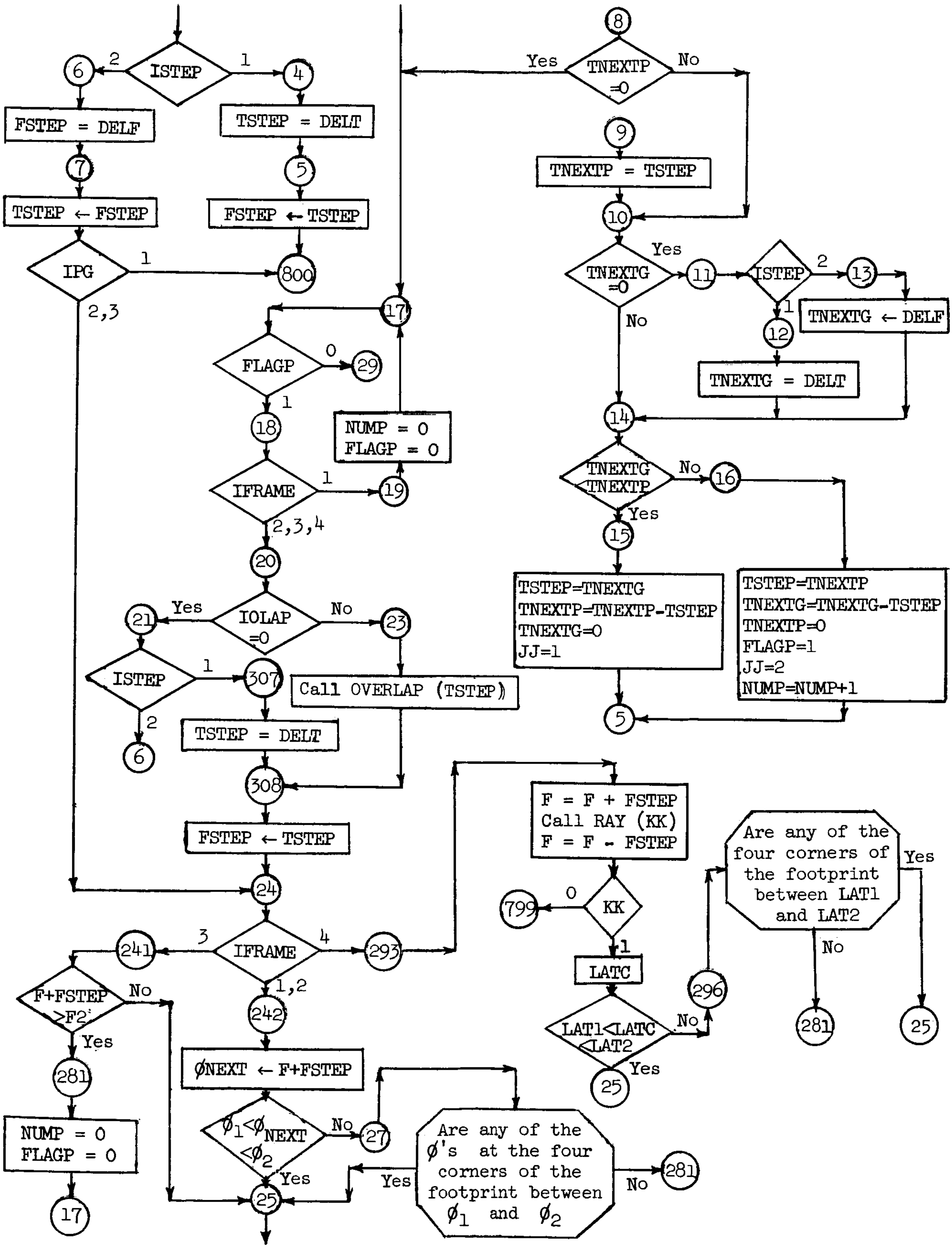
Flow Diagram for
Subroutine LOGIC

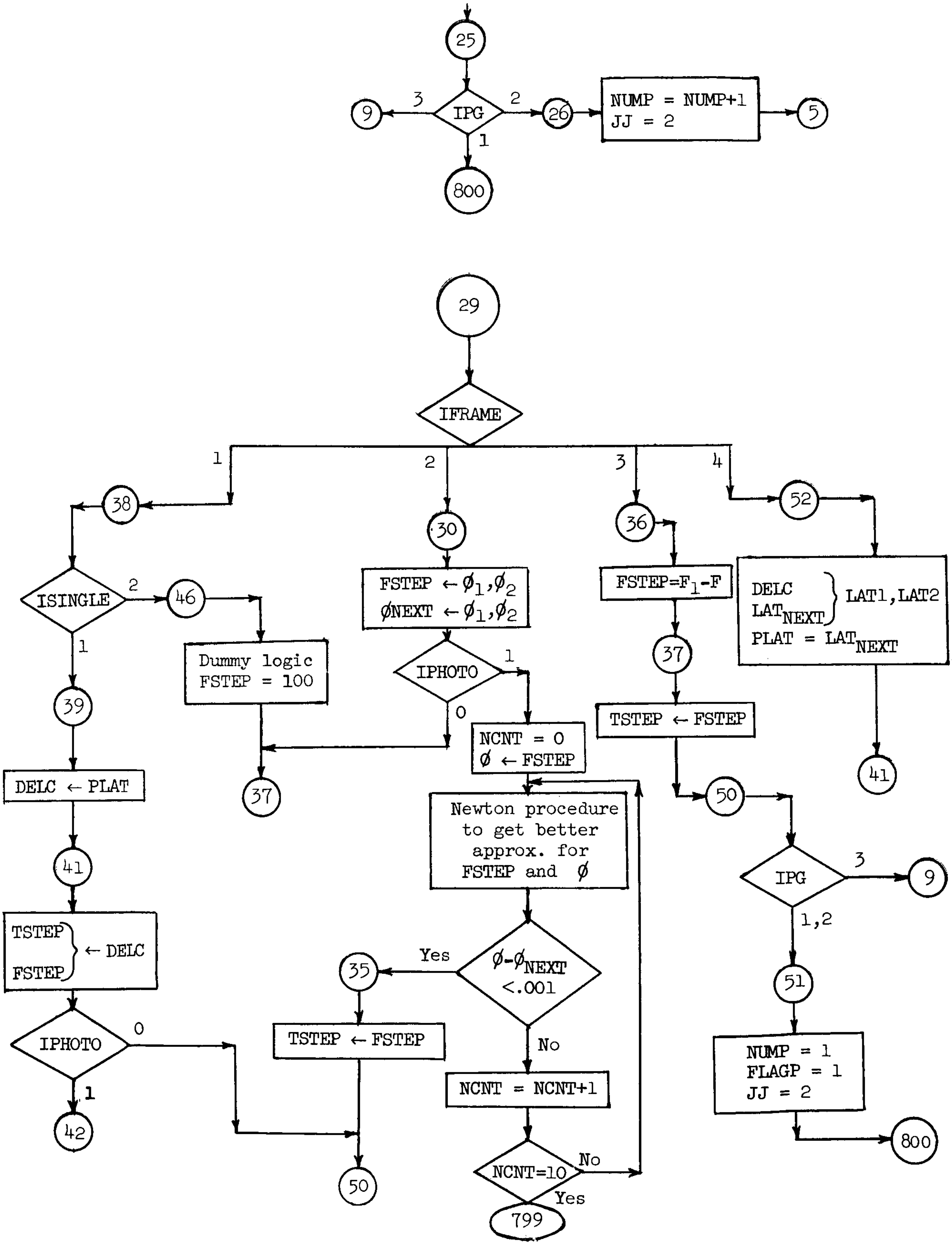


APPENDIX A

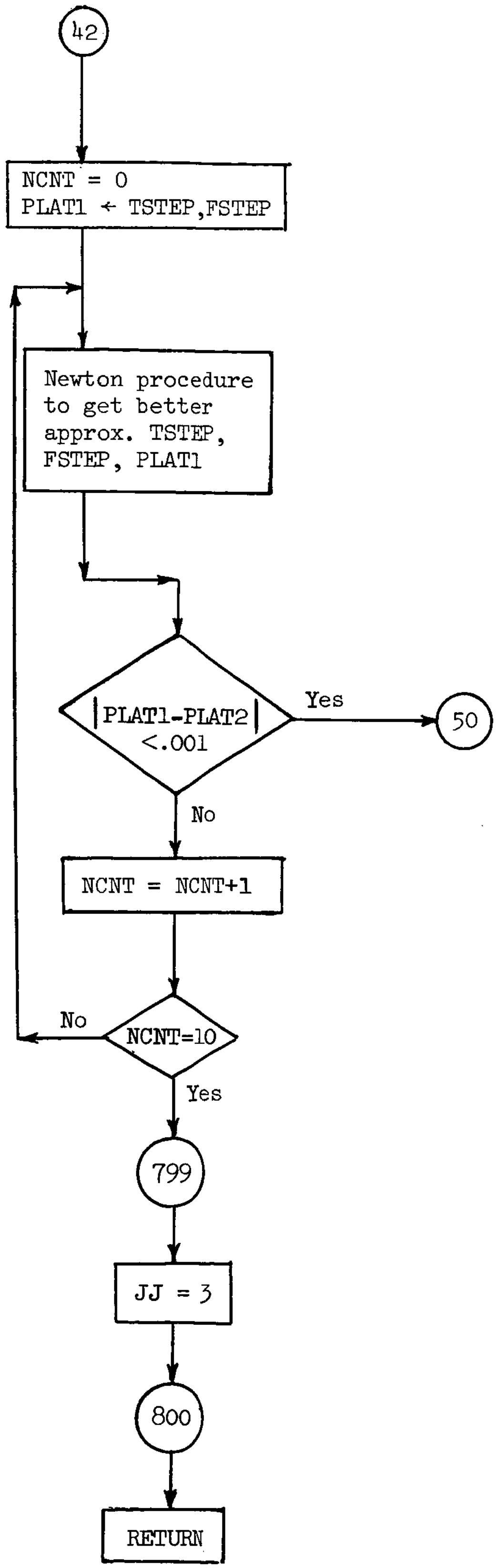

102
IFRAME
1
2
3
4
106
103
105
108
1,2
IPG
2
3
107
Call GPRINT
2
Call RAY(KK)
KK
0
2
1
Call DOT (Ø)
No
Ø1 < Ø < Ø2
Yes
F1<F<F2
Yes
No
106
Call RAY (KK)
KK
0
2
1
LATC
LAT1<LATC<LAT2
No
106
Yes
104
FLAGP = 1
NUMP = 1
Call PPRINT
2
JJ=1
3
1
IPG
3
8
2

APPENDIX A


ISTEP
FSTEP = DELF
TSTEP = DELT
TSTEP ← FSTEP
FSTEP ← TSTEP
IPG
TNEXTP =0
TNEXTP = TSTEP
TNEXTG =0
TNEXTG ← DELF
TNEXTG = DELT
FLAGP
NUMP = 0
FLAGP = 0
IFRAME
TNEXTG <TNEXTP
TSTEP=TNEXTG
TNEXTP=TNEXTP-TSTEP
TNEXTG=0
JJ=1
TSTEP=TNEXTP
TNEXTG=TNEXTG-TSTEP
TNEXTP=0
FLAGP=1
JJ=2
NUMP=NUMP+1
IOLAP =0
Call OVERLAP (TSTEP)
TSTEP = DELT
FSTEP ← TSTEP
F = F + FSTEP
Call RAY (KK)
F = F - FSTEP
KK
LATC
LAT1<LATC <LAT2
Are any of the four corners of the footprint between LAT1 and LAT2
F+FSTEP >F2
ØNEXT ← F+FSTEP
Ø1<ØNEXT <Ø2
Are any of the Ø's at the four corners of the footprint between Ø1 and Ø2
Yes
No


25
IPG
3
2
1
9
26
NUMP = NUMP+1
JJ = 2
5
800
29
IFRAME
1
2
3
4
38
ISINGLE
2
46
Dummy logic
FSTEP = 100
1
39
DELC ← PLAT
41
TSTEP
FSTEP
← DELC
IPHOTO
0
1
42
37
30
FSTEP ← Ø1,Ø2
ØNEXT ← Ø1,Ø2
IPHOTO
1
0
NCNT = 0
Ø ← FSTEP
Newton procedure
to get better
approx. for
FSTEP and Ø
Ø-ØNEXT
<.001
Yes
35
TSTEP ← FSTEP
50
No
NCNT = NCNT+1
NCNT=10
No
Yes
799
36
FSTEP=F1-F
37
TSTEP ← FSTEP
50
IPG
3
9
1,2
51
NUMP = 1
FLAGP = 1
JJ = 2
800
52
DELC
LATNEXT
LAT1,LAT2
PLAT = LATNEXT
41

APPENDIX A



42
NCNT = 0
PLAT1 ← TSTEP,FSTEP
Newton procedure to get better approx. TSTEP, FSTEP, PLAT1
|PLAT1-PLAT2| <.001
Yes
50
No
NCNT = NCNT+1
No
NCNT=10
Yes
799
JJ = 3
800
RETURN

APPENDIX B

PROGRAM LISTING

Program PANDG contains a main program and 31 subroutines. A FORTRAN IV listing of the program is contained in this appendix. The purpose of each subroutine is stated in the listing. Several of the subroutines contained in PANDG are not presented herein, since their listings appear in reference 1. The following subroutine listings have beem omitted: CARSPH, CONCAR, DOT, EEARTH, EMARS, EULER, JULCAL, LATLNG, ORBIT, PRECES, QADRAT, RECEQ, REQMEQ, REQPEQ, RXYZPQW, SUNBAND, TCONIC, TINVS, VECTOR.

APPENDIX B

```
      PROGRAM PANDG     (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
      COMMON/BLOCK1/JDO,JD,DATE,ONUM,HA,OLAST,DELT,DELF
      COMMON/BLOCK2/A,E,XI,W,O,F,PERIOD,ANGRAT,FO,FTOTAL,TTOTAL
      COMMON/BLOCK3/RD,DR,U,RS,PI,HAREF,HADOT,JDREF
      COMMON/BLOCK4/SX,SY,SZ,EX,EY,EZ,CX,CY,CZ
      COMMON/BLOCK5/IPG,IPUT,IDATE,ISTEP,IOCC,IPHOTO,IMANU,IB,IOLAP,IFRA
     *ME,ISINGLE
      COMMON/BLOCK6/TSTOP,FSTOP,OSTOP
      COMMON/BLOCK7/HAO,HPO,AO,EO,XIO,WO,OO,VINF,SDEC,SRA,BETA
      COMMON/BLOCK8/XJ20,RES,PHI1,PHI2,OVLAP,CANGF,CANGS,FPHOTO1,FPHOTO2
     *,FOOTL,LAMBDA,LAT1,LAT2,PLAT
      DIMENSION DATE(6),FOOTL(3,4)
      REAL LAT1,LAT2,LAMBDA,LATLAST
      REAL JD,JDO,JDREF
      INTEGER OSTOP,ONUM,OLAST
      NAMELIST/CASE/HAO,HPO,AO,EO,XIO,WO,OO,FO,DATE,XJ20,RES,VINF,
     *SDEC,SRA,BETA,IPG,IPUT,IDATE,ISTEP,IOCC,IPHOTO,IMANU,IB,IOLAP,IFRA
     *ME,FSTOP,OSTOP,TSTOP,DELT,DELF,RS,U,PHI1,PHI2,JD,OVLAP,CANGF,CANGS
     *,FPHOTO1,FPHOTO2,LAT1,LAT2,PLAT,LAMBDA,HAREF,HADOT,JDREF,ISINGLE
      ANGLE(X)=AMOD(X,360.)+180.-SIGN(180.,X)
C
    1 CONTINUE
C
      CALL STDCASE
C
      READ(5,CASE)
C
      GO TO (2,3,4),IPUT
C
    2 A=AO $ E=EO $ XI=XIO $ W=WO $ O=OO $ F=FO
      GO TO 5
C
    3 AO=(HPO+HAO+2.*RS)/2.
      EO=1.-(RS+HPO)/AO
      GO TO 2
C
    4 CALL ORBIT(SDEC,SRA,VINF,HAO,HPO,BETA,A,E,XI,W,O,PDEC,PRA,U,RS)
      F=0. $ FO=0.
    5 IF(A.LT.0) GO TO 1
      PERIOD=2.*PI*SQRT(A**3/U)
      ANGRAT=2.*PI/PERIOD
```

PANDG

```
C
    6 GO TO(7,8),IDATE
C
    7 CALL CALJUL(WJD,FJD,WND,FD,DATE)
      JD=WJD+FJD
      GO TO 9
C
    8 CONTINUE
      WJD=FLOAT(IFIX(JD))
      FJD=JD-WJD
      CALL JULCAL(DATE,WJD,FJD,0)
C
    9 JDO=JD
      HA=ANGLE(HAREF+HADOT*(JD-JDREF))
C
      CALL VECTOR(JD,DECS,RAS,DECE,RAE,DECC,RAC,SX,SY,SZ,EX,EY,EZ,CX,CY,
     *CZ,4)
C
      CALL OPRINT(0)
      CALL LOGIC(0,TSTEP,FSTEP,JJ)
   10 CALL LOGIC(1,TSTEP,FSTEP,JJ)
C
      CALL UPDATE(TSTEP,FSTEP)
C
      IF((JD-JDO)*86400.GT.TSTOP) GO TO 1
      IF(ONUM.GT.OSTOP) GO TO 1
      IF(FTOTAL.GT.FSTOP) GO TO 1
C
      IF(OLAST.NE.ONUM) CALL OPRINT(1)
C
      GO TO (11,12,13),JJ
C
   11 CALL GPRINT
      GO TO 10
C
   12 CALL PPRINT
      GO TO 10
C
   13 WRITE(6,101)
  101 FORMAT(*0NO SOLUTION FOUND *)
      GO TO 1
      END
```

APPENDIX B

```
      SUBROUTINE PRATIO(FOOTO,FOOTF,RATIO)
C
C     THIS SUBROUTINE CALCULATES THE OVERLAP RATIO BETWEEN TWO FOOTPRINTS
C
C     FOOTO - MATRIX OF 4 COLUMN VECTORS DEFINING PREVIOUS FOOTPRINT
C     FOOTF - MATRIX OF 4 COLUMN VECTORS DEFINING PRESENT FOOTPRINT
C     RATIO - FOOTPRINT OVERLAP RATIO
C
      COMMON/BLOCK3/RD,DR,U,RS,PI,HAREF,HADCT,JDREF
      COMMON/BLOCK5/IPG,IPUT,IDATE,ISTEP,IOCC,IPHOTO,IMANU,IB,IOLAP,
     1IFRAME,ISINGLE
C
      REAL MAGV,NORM,NORMO,NORMF,NFJOI,NFLOI,MAG,MAGJI,MAGLI,NOJFI,NOLFI
C
      DIMENSION FOOT(3,6),FOOTO(3,4),FOOTF(3,4),NORM(3,6),NORMO(3,4),NOR
     *MF(3,4),VINT(3),ANGO(4),NFJOI(3),NFLOI(3),ANGF(4),NOJFI(3),NOLFI(3
     *),CANO(4),CANF(4)
      DIMENSION C(3),X(3),AZ(8),ISTORE(8)
      ANGLE(X)=AMOD(X,360.)+180.-SIGN(180.,X)
C
      RATIO=0. $ EXCES=0.
C
      N=0
C
      DO 3 I=1,4
      J=I+1 $ IF(J.EQ.5) J=1
      NORMO(1,I)=FOOTO(2,I)*FOOTO(3,J)-FOOTO(2,J)*FOOTO(3,I)
      NORMO(2,I)=FOOTO(1,J)*FOOTO(3,I)-FOOTO(1,I)*FOOTO(3,J)
      NORMO(3,I)=FOOTO(1,I)*FOOTO(2,J)-FOOTO(1,J)*FOOTO(2,I)
      NORMF(1,I)=FOOTF(2,I)*FOOTF(3,J)-FOOTF(2,J)*FOOTF(3,I)
      NORMF(2,I)=FOOTF(1,J)*FOOTF(3,I)-FOOTF(1,I)*FOOTF(3,J)
      NORMF(3,I)=FOOTF(1,I)*FOOTF(2,J)-FOOTF(1,J)*FOOTF(2,I)
C
      TEMPO=0. $ TEMPF=0.
      DO 1 K=1,3
      TEMPO=TEMPO+NORMO(K,I)**2
    1 TEMPF=TEMPF+NORMF(K,I)**2
      TEMPO=SQRT(TEMPO)
      TEMPF=SQRT(TEMPF)
      IF(ABS(TEMPO).LT..01)GO TO 800
      IF(ABS(TEMPF).LT..01)GO TO 800
C
```

PRATIO

```
      CO 2 K=1,3
      NORMO(K,I)=NCRMO(K,I)/TEMPC
    2 NORMF(K,I)=NCRMF(K,I)/TEMPF
C
      ANGO(I)=ACOS((FOOTO(1,I)*FCOTO(1,J)+FCOTO(2,I)*FOOTO(2,J)+FOOTC(3,
     *I)*FOOTO(3,J))/RS/RS)*RC
    3 ANGF(I)=ACOS((FOCTF(1,I)*FCOTF(1,J)+FOOTF(2,I)*FOOTF(2,J)+FOCTF(3,
     *I)*FCCTF(3,J))/RS/RS)*RC
C
      ANG=0. $ SUM=0.
      J=4
      DO 4 I=1,4
      CANO(I)=ACOS(-1.*(NORMC(1,J)*NCFMC(1,I)+NORMO(2,J)*NORMO(2,I)+NCRM
     *O(3,J)*NORMO(3,I)))*RD
      ANG=ANG+CANO(I)
      CANF(I)=ACOS(-1.*(NORMF(I,J)*NCRMF(1,I)+NORMF(2,J)*NOFMF(2,I)+NCRM
     *F(3,J)*NCRMF(3,I)))*RD
      SUM=SUM+CANF(I)
    4 J=I
C
      EXCESC=ANG-360.
      EXCESF=SUM-36C.
      M1=-1$M2=1$M3=2
C
      DO 5 I=1,4
      K=I+1 $ IF(K.EQ.5) K=1
      DO 5 J=1,4
      L=J+1 $ IF(L.EQ.5) L=1
C
      DO 5 ISIGN=M1,M2,M3
      VINT(1)=NCRMC(2,I)*NCRMF(3,J)-NCFMF(2,J)*NORMC(3,I)
      VINT(2)=NCRMF(1,J)*NORMO(3,I)-NCRMO(1,I)*NORMF(3,J)
      VINT(3)=NORMO(1,I)*NORMF(2,J)-NCFMF(1,J)*NCRMC(2,I)
      MAGV=SQRT(VINT(1)**2+VINT(2)**2+VINT(3)**2)*ISIGN
      VINT(1)=VINT(1)/MAGV $ VINT(2)=VINT(2)/MAGV $ VINT(3)=VINT(3)/MAGV
C
      ANGVI=ACOS(    (FCCTO(1,I)*VINT(1)+FOOTO(2,I)*VINT(2)+FOOTO(3,I)*VI
     *NT(3))/RS)*RD
      ANGVK=ACOS(    (FOOTO(1,K)*VINT(1)+FOOTO(2,K)*VINT(2)+FOOTO(3,K)*VI
     *NT(3))/RS)*RD
C
      IF(ABS(ANGVI+ANGVK-ANGC(I)).GT..CO1) GC TO 5
C
```

APPENDIX B

PRATIO

```
      ANGVJ=ACOS(    (FOOTF(1,J)*VINT(1)+FOOTF(2,J)*VINT(2)+FOOTF(3,J)*VI
     *NT(3))/RS)*RD
      ANGVL=ACOS(    (FOOTF(1,L)*VINT(1)+FOOTF(2,L)*VINT(2)+FOOTF(3,L)*VI
     *NT(3))/RS)*RD
C
      IF(ABS(ANGVJ+ANGVL-ANGF(J)).GT.1.E-6) GO TO 5
C
      TEMP=VINT(1)*FOOTO(1,I)+VINT(2)*FOOTO(2,I)+VINT(3)*FOOTO(3,I)
      IF(TEMP.LT.0.) 41,42
   41 VINT(1)=-VINT(1) $ VINT(2)=-VINT(2) $ VINT(3)=-VINT(3)
C
   42 N=N+1
      FOOT(1,N) =VINT(1)
      FOOT(2,N) =VINT(2)
      FOOT(3,N) =VINT(3)
C
    5 CONTINUE
C
      DO 8 I=1,4
      SUM=0.
      DO 6 J=1,4
      L=J+1 $ IF(L.EQ.5) L=1
C
      NFJOI(1)=FOOTF(2,J)*FOOTO(3,I)-FOOTO(2,I)*FOOTF(3,J)
      NFJOI(2)=FOOTO(1,I)*FOOTF(3,J)-FOOTF(1,J)*FOOTO(3,I)
      NFJOI(3)=FOOTF(1,J)*FOOTO(2,I)-FOOTO(1,I)*FOOTF(2,J)
      MAGJI=SQRT(NFJOI(1)**2+NFJOI(2)**2+NFJOI(3)**2)
      NFLOI(1)=FOOTF(2,L)*FOOTO(3,I)-FOOTO(2,I)*FOOTF(3,L)
      NFLOI(2)=FOOTO(1,I)*FOOTF(3,L)-FOOTF(1,L)*FOOTO(3,I)
      NFLOI(3)=FOOTF(1,L)*FOOTO(2,I)-FOOTO(1,I)*FOOTF(2,L)
      MAGLI=SQRT(NFLOI(1)**2+NFLOI(2)**2+NFLOI(3)**2)
C
    6 SUM=SUM+ACOS(    (NFJOI(1)*NFLOI(1)+NFJOI(2)*NFLOI(2)+NFJOI(3)*NFLO
     *    I(3))/MAGJI/MAGLI)*RD+ACOS(    (NFJOI(1)*NORMF(1,J)+NFJOI(2)*NO
     *    RMF(2,J)+NFJOI(3)*NORMF(3,J))/MAGJI)*RD+ACOS(-1*(NFLOI(1)*NORM
     *    F(1,J)+NFLOI(2)*NORMF(2,J)+NFLOI(3)*NORMF(3,J))/MAGLI)*RD-180.
C
      IF(ABS(SUM-EXCESF).LT..001) 7,8
C
    7 N=N+1
      FOOT(1,N)=FOOTO(1,I)
      FOOT(2,N)=FOOTO(2,I)
      FOOT(3,N)=FOOTO(3,I)
```

PRATIO

```
      THETAI=CANO(I)
C
    8 CONTINUE
C
      DO 11  I=1,4
      SUM=0.
      DO   9  J=1,4
      L=J+1  $  IF(L.EQ.5)  L=1
C
      NOJFI(1)=FOOTO(2,J)*FOOTF(3,I)-FOOTF(2,I)*FOOTO(3,J)
      NOJFI(2)=FOOTF(1,I)*FOOTO(3,J)-FOOTO(1,J)*FOOTF(3,I)
      NOJFI(3)=FOOTO(1,J)*FOOTF(2,I)-FOOTF(1,I)*FOOTO(2,J)
      MAGJI=SQRT(NOJFI(1)**2+NOJFI(2)**2+NOJFI(3)**2)
      NOLFI(1)=FOOTO(2,L)*FOOTF(3,I)-FOOTF(2,I)*FOOTO(3,L)
      NOLFI(2)=FOOTF(1,I)*FOOTO(3,L)-FOOTO(1,L)*FOOTF(3,I)
      NOLFI(3)=FOOTO(1,L)*FOOTF(2,I)-FOOTF(1,I)*FOOTO(2,L)
      MAGLI=SQRT(NOLFI(1)**2+NOLFI(2)**2+NOLFI(3)**2)
C
    9 SUM=SUM+ACOS(    (NOJFI(1)*NOLFI(1)+NOJFI(2)*NOLFI(2)+NOJFI(3)*NOLF
     *    I(3))/MAGJI/MAGLI)*RD+ACOS(    (NOJFI(1)*NORMO(1,J)+NOJFI(2)*NO
     *    RMO(2,J)+NOJFI(3)*NORMO(3,J))/MAGJI)*RD+ACOS(-1*(NOLFI(1)*NORM
     *    O(1,J)+NOLFI(2)*NORMO(2,J)+NOLFI(3)*NORMO(3,J))/MAGLI)*RD-180.
C
      IF(ABS(SUM-EXCESO).LT.1.E-6)  10,11
C
   10 N=N+1
      FOOT(1,N)=FOOTF(1,I)
      FOOT(2,N)=FOOTF(2,I)
      FOOT(3,N)=FOOTF(3,I)
      THETAI=CANF(I)
C
   11 CONTINUE
C
      IF(N.EQ.0)  GO TO 800
  209 NC=N
      DO 201  I=1,NC
      MAG=SQRT(FOOT(1,I)**2+FOOT(2,I)**2+FOOT(3,I)**2)
      DO 201  J=1,3
  201 FOOT(J,I)=FOOT(J,I)/MAG
C
      DO 202  I=1,3
  202 C(I)=(FOOT(I,1)+FOOT(I,2))/2.
C
```

PRATIO

```
      MAG=SQRT(C(1)**2+C(2)**2+C(3)**2)
      DO 203 I=1,3
  203 C(I)=C(I)/MAG
C
      XY=SQRT(C(1)**2+C(2)**2)
      ST=C(2)/XY
      CT=C(1)/XY
      SP=C(3)
      CP=SQRT(1.-SP*SP)
      DO 205 I=1,NC
      DO 204 J=1,3
  204 X(J)=FOOT(J,I)-C(J)
C
      YP=-X(1)*ST+X(2)*CT
      ZP=-X(1)*SP*CT-X(2)*SP*ST+X(3)*CP
      AZ(I)=ANGLE(ATAN2(YP,ZP)*RD)
  205 ISTORE(I)=I
C
      DO 206 I=1,NC
      M=NC-I
      DO 206 J=1,M
      IF(AZ(J).LT.AZ(J+1)) GO TO 206
      ATEMP=AZ(J)
      ITEMP=ISTORE(J)
      AZ(J)=AZ(J+1)
      ISTORE(J)=ISTORE(J+1)
      AZ(J+1)=ATEMP
      ISTORE(J+1)=ITEMP
  206 CONTINUE
C
      DO 207 K=1,NC
      I=ISTORE(K)
      J=ISTORE(K+1)
      IF(K.EQ.NC) J=ISTORE(1)
      NORM(1,K)=FOOT(2,I)*FOOT(3,J)-FOOT(2,J)*FOOT(3,I)
      NORM(2,K)=FOOT(1,J)*FOOT(3,I)-FOOT(1,I)*FOOT(3,J)
      NORM(3,K)=FOOT(1,I)*FOOT(2,J)-FOOT(1,J)*FOOT(2,I)
      MAG=SQRT(NORM(1,K)**2+NORM(2,K)**2+NORM(3,K)**2)
      DO 207 L=1,3
  207 NORM(L,K)=NORM(L,K)/MAG
C
      J=NC $ SUM=0.
      DO 208 I=1,NC
      SUM=SUM+ACOS(-1.*(NORM(1,I)*NORM(1,J)+NORM(2,I)*NORM(2,J)+NORM(3,I
     *)*NORM(3,J)))*RD
  208 J=I
C
      EXCES=SUM-(NC-2)*180.
      RATIO=EXCES/EXCESO
C
  800 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE LOGIC(K,TSTEP,FSTEP,JJ)
C
C     THIS SUBROUTINE EXAMINES PROGRAM OPTIONS TO DETERMINE WHETHER THE
C     NEXT EVENT IS A GROUNDTRACK OR PHOTOGRAPHY EVENT AND CALCULATES
C     THE TIME INCREMENT AND TRUE ANOMALY INCREMENT TO THE NEXT EVENT
C
C     K=0   INITIALIZE
C     K=1   COMPUTE NEXT STEP ( TSTEP IN SEC AND FSTEP IN DEG )
C     TSTEP - TIME INCREMENT TO NEXT EVENT
C     FSTEP - TRUE ANOMALY INCREMENT TO NEXT EVENT
C     JJ=1 UPDATE AND CALL SUB GPRINT
C     JJ=2 UPDATE AND CALL SUB PPRINT
C     JJ=3 NO SOLUTION
C
      COMMON/BLOCK1/JDO,JD,DATE,ONUM,FA,OLAST,DELT,DELF
      COMMON/BLOCK2/A,E,XI,W,O,F,PERIOD,ANGRAT,FO,FTOTAL,TTOTAL
      COMMON/BLOCK3/RD,DR,U,RS,PI,HAREF,HADOT,JDREF
      COMMON/BLOCK4/SX,SY,SZ,EX,EY,EZ,CX,CY,CZ
      COMMON/BLOCK5/IPG,IPUT,IDATE,ISTEP,IOCC,IPHOTO,IMANU,IB,IOLAP,IFRA
     *ME,ISINGLE
      COMMON/BLOCK6/TSTOP,FSTOP,OSTOP
      COMMON/BLOCK7/HAO,HPO,AO,EO,XIO,WO,OO,VINF,SDEC,SRA,BETA
      COMMON/BLOCK8/XJ20,RES,PHI1,PHI2,OVLAP,CANGF,CANGS,FPHOTO1,FPHOTO2
     *,FOOTL,LAMBDA,LAT1,LAT2,PLAT
      COMMON/BLOCK9/NUMP
      REAL LAT1,LAT2,LAMBDA,LATLAST
      REAL M2,JDTEMP
      REAL JD,JDO,JDREF
      INTEGER OSTOP,ONUM,OLAST
      INTEGER FLAGP
      DIMENSION DATE(6),FOOTL(3,4)
      DIMENSION RPOW(3,3)
      ANGLE(X)=AMOD(X,360.)+180.-SIGN(180.,X)
C
C
      KK=K+1
      GO TO (1,101),KK
C
    1 FLAGP=0
      TNEXTP=0.
      TNEXTG=0.
      NUMP=0
```

APPENDIX B

LOGIC

```
      GO TO 8CC
C
  101 IF(ABS(FTCTAL).GT..001) GC TC 2
      GO TO (1C7,102,1C2),IPG
  102 GO TO (1C6,103,1C5,108),IFRAME
  103 CALL RAY(C.,0.,XC,YC,ZC,KK)
      IF(KK.EQ.C) GO TO 2
      CALL DCT(SX,SY,SZ,XC,YC,ZC,G)
      IF(G.LT.PHI2.AND.G.GT.PHI1) 104,106
  104 FLAGP=1
      NUMP=1
      CALL PPRINT
      GO TO 2
  105 DELF12=ANGLE(FPHCTC2-FPHCTC1)
      DELFO2=ANGLE(FPHOTO2-F)
      IF(DELFO2.LT.DELF12) 104,1C6
  106 GO TO (2,2,107),IPG
C
  108 CALL RAY(C.,0.,XC,YC,ZC,KK)
      IF(KK.EQ.0) GO TO 2
      SINC=ZC/RS
      IF(LAT1.GT.LAT2) 1C9,110
  109 TEMP=LAT2
      LAT2=LAT1
      LAT1=TEMP
  110 SIN1=SIN(LAT1*DR)
      SIN2=SIN(LAT2*DR)
      KK=1
      IF(SINC.LT.SIN2.AND.SINC.GT.SIN1) KK=2
      GO TO (106,104),KK
  107 CALL GPRINT
C
    2 GO TO (3,17,8),IPG
    3 JJ=1
      GO TO (4,6),ISTEP
C
    4 TSTEP=DELT
    5 CALL TCONIC(U,E,A,F,T1)
      M2=ANGRAT*(T1+TSTEP)
      CALL TINVS(M2,E,EC2,F2)
      F2=ANGLE(F2*RD)
      FSTEP=F2-F
      IF(FSTEP.LT.C.) FSTEP=36C.+FSTEP
```

LOGIC

```
      GO TO 800
C
    6 FSTEP=DELF
    7 CALL TCONIC(U,E,A,F,T1)
      CALL TCONIC(U,E,A,F+FSTEP,T2)
      TSTEP=T2-T1
      IF(TSTEP.LT.0.) TSTEP=PERIOD+TSTEP
      GO TO (800,24,24),IPG
C
    8 CONTINUE
C
      IF(ABS(TNEXTP).LT.1.E-6) 17,10
    9 TNEXTP=TSTEP
C
   10 IF(ABS(TNEXTG).LT.1.E-6) 11,14
C
   11 GO TO (12,13),ISTEP
C
   12 TNEXTG=DELT
      GO TO 14
C
   13 CALL TCONIC(U,E,A,F,T1)
      CALL TCONIC(U,E,A,F+DELF,T2)
      TNEXTG=T2-T1
      IF(TNEXTG.LT.0.) TNEXTG=PERIOD+TNEXTG
C
   14 IF(TNEXTG.LT.TNEXTP) 15,16
C
   15 TSTEP=TNEXTG
      TNEXTP=TNEXTP-TSTEP
      TNEXTG=0.
      JJ=1
      GO TO 5
C
   16 TSTEP=TNEXTP
      TNEXTG=TNEXTG-TSTEP
      TNEXTP=0.
      FLAGP=1
      NUMP=NUMP+1
      JJ=2
      GO TO 5
C
   17 CONTINUE
```

LOGIC

```
      IF(FLAGP.EQ.0) 29,18
C
   18 GO TO (19,20,20,20),IFRAME
   19 FLAGP=0
      NUMP=0
      GO TO 17
C
   20 IF(IOLAP.EQ.0) 21,23
C
   21 GO TO (307,6),ISTEP
C
   23 CALL OVERLAP(TSTEP)
      GO TO 308
  307 TSTEP=DELT
  308 CALL TCONIC(U,E,A,F,T1)
      M2=ANGRAT*(T1+TSTEP)
      CALL TINVS(M2,E,EC2,F2)
      F2=ANGLE(F2*RD)
      FSTEP=F2-F
      IF(FSTEP.LT.C.)FSTEP=360.+FSTEP
C
   24 GO TO (242,242,241,293),IFRAME
C
  241 CALL TCONIC(U,E,A,F,T1)
      M2=ANGRAT*(T1+TSTEP)
      CALL TINVS(M2,E,EC2,FNEXT)
      FNEXT=ANGLE(FNEXT*RD)
      IF(FNEXT.GT.FPHOTO2) 281,25
C
  242 JDTEMP=JD+TSTEP/86400.
      CALL VECTOR(JDTEMP,B1,B2,B3,B4,B5,B6,SXTEMP,SYTEMP,SZTEMP,B7,B8,B9
     *,B10,B11,B12,4)
      F=F+FSTEP
      CALL RAY(0.,0.,XC,YC,ZC,KK)
      F=F-FSTEP
      CALL DOT(SXTEMP,SYTEMP,SZTEMP,XC,YC,ZC,GTEMP)
C
      IF(GTEMP.LT.PHI2.AND.GTEMP.GT.PHI1) 25,27
C
   25 GO TO (800,26,9),IPG
   26 JJ=2
      NUMP=NUMP+1
      GO TO 5
```

LOGIC

```
C
   27 CONTINUE
      DO 28 I=1,2
      DO 28 J=1,2
      F=F+FSTEP
      CALL RAY((-1.)**I*CANGF,(-1.)**J*CANGS,XC,YC,ZC,KK)
      F=F-FSTEP
      IF(KK.EQ.0) GO TO 28
      CALL DCT(SXTEMP,SYTEMP,SZTEMP,XC,YC,ZC,GTEMP)
      IF(GTEMP.LT.PHI2.AND.GTEMP.GT.PHI1) GO TO 25
   28 CONTINUE
C
  281 FLAGP=0
      NUMP=0
      GO TO 17
C
  293 F=F+FSTEP
      CALL RAY(C.,C.,XT,YT,ZT,KK)
      F=F-FSTEP
      IF(KK.EQ.0) GO TO 799
      SINC=ZT/RS
      PLATC=ASIN(SINC)*RD
      IF(SINC.GT.SIN2.OR.SINC.LT.SIN1) GO TO 296
  294 GO TO 25
C
  296 F=F+FSTEP
      IABOVE=0
      IBELOW=0
      DO 299 I=1,2
      DO 299 J=1,2
      CALL RAY((-1.)**I*CANGF,(-1.)**J*CANGS,XT,YT,ZT,KK)
      IF(KK.EQ.0) GO TO 299
      SLAT=ZT/RS
      PLAT1=ASIN(SLAT)*RD
      IF(SLAT.GT.SIN2) IABOVE=IABOVE+1
      IF(SLAT.LT.SIN1) IBELOW=IBELOW+1
  299 CONTINUE
      IF(IABOVE.EQ.4.OR.IBELOW.EQ.4) 300,305
  300 CONTINUE
      F=F-FSTEP
      GO TO 281
  305 F=F-FSTEP
      GO TO 25
```

APPENDIX B

LOGIC

```
C
   29 CONTINUE
C
      GO TO (38,30,36,52),IFRAME
C
   30 CALL  SUNBAND(SX,SY,SZ,PHI1,XI,W,C,RPQW,F1,F2,ITYPE1,ITYPE2,0)
      CALL  SUNBAND(SX,SY,SZ,PHI2,XI,W,0,RPQW,F3,F4,ITYPE3,ITYPE4,0)
      IF(ITYPE1.EQ.0) F1=1000.
      IF(ITYPE2.EQ.0) F2=1000.
      IF(ITYPE3.EQ.0) F3=1000.
      IF(ITYPE4.EQ.0) F4=1000.
      IF(AMIN1(F1,F2,F3,F4).GT.999.) GO TO 799
      DF=AMIN1(ANGLE(F1-F),ANGLE(F2-F),ANGLE(F3-F),ANGLE(F4-F))
      FSTEP=DF
      IF(ABS(DF-ANGLE(F1-F)).LT..001) GO TO 31
      IF(ABS(DF-ANGLE(F2-F)).LT..001) GO TO 31
      GNEXT=PHI2
      GO TO 32
   31 GNEXT=PHI1
   32 CONTINUE
C
      IF(IPHOTO.EQ.0) GO TO 37
C
      NCNT=0
      CALL  TCONIC(U,E,A,F,T1)
      CALL  TCONIC(U,E,A,F+DF,T2)
      DT=T2-T1
      IF(DT.LT.0.) DT=PERIOD+DT
C
      CALL  UPDATE(DT,DF)
      T1=T2
      CALL  RAY(0.,0.,XC,YC,ZC,KK)
      CALL  DCT(SX,SY,SZ,XC,YC,ZC,G1)
C
   33 CONTINUE
      CALL  TCONIC(U,E,A,F+1.,T2)
      TSTEP=T2-T1
      IF(TSTEP.LT.0.) TSTEP=PERIOD+TSTEP
      DF=DF+1.
      DT=DT+TSTEP
      CALL  UPDATE(TSTEP,1.)
      CALL  RAY(0.,0.,XC,YC,ZC,KK)
      CALL  DCT(SX,SY,SZ,XC,YC,ZC,G2)
```

LOGIC

```
      PGWRTF=(G2-G1)/1.
      FSTEP=-(G1-GNEXT)/PGWRTF-1.
      T1=T2
      CALL TCONIC(U,E,A,F+FSTEP,T2)
      TSTEP=T2-T1
      IF(FSTEP.GT.0..AND.TSTEP.LT.0.) TSTEP=PERIOD+TSTEP
      IF(FSTEP.LT.0..AND.TSTEP.GT.0.) TSTEP=TSTEP-PERIOD
      DF=DF+FSTEP
      DT=DT+TSTEP
      CALL UPDATE(TSTEP,FSTEP)
      CALL RAY(0.,0.,XC,YC,ZC,KK)
      CALL DOT(SX,SY,SZ,XC,YC,ZC,G1)
      T1=T2
      IF(ABS(G1-GNEXT).LT..001) 35,34
C
   34 NCNT=NCNT+1
      IF(NCNT.EQ.10) 799,33
C
   35 CONTINUE
      TSTEP=DT
      FSTEP=DF
      CALL UPDATE(-DT,-DF)
      GO TO 50
C
   36 CONTINUE
      FSTEP=ANGLE(FPHOTO1-F)
C
   37 CALL TCONIC(U,E,A,F,T1)
      CALL TCONIC(U,E,A,F+FSTEP,T2)
      TSTEP=T2-T1
      IF(TSTEP.LT.0.) TSTEP=TSTEP+PERIOD
      GO TO 50
C
   38 GO TO (39,46),ISINGLE
   39 SC1=SIN(PLAT*DR)/SIN(XI*DR)
      IF(ABS(SC1).GE.1.) GO TO 799
      C1=ASIN(SC1)*RD
      C2=SIGN(180.,C1)-C1
      CO=ANGLE(W+F)
      IF(ANGLE(C1-CO).LT..001)C1=CO-.1
      IF(ANGLE(C2-CO).LT..001)C2=CO-.1
      DELC=AMIN1(ANGLE(C1-CO),ANGLE(C2-CO))
   41 CALL TCONIC(U,E,A,F,T1)
```

APPENDIX B

LOGIC

```
      CALL TCONIC(U,E,A,F+DELC,T2)
      DT=T2-T1
      IF(DT.LT.0.) DT=PERIOD+DT
      OSTAR=0
      WSTAR=W
      CALL WELLS(RS,U,XJ20,A,E,XI,WSTAR,OSTAR,DT)
      FL=ANGLE(CC+DELC-WSTAR)
      FSTEP=ANGLE(FL-F)
      TSTEP=DT
      IF(IPHOTO.EQ.0) GO TO 50
C
   42 NCNT=0
      CALL UPDATE(TSTEP,FSTEP)
      CALL RAY(0.,0.,XT,YT,ZT,KK)
      IF(KK.EQ.0) GO TO 799
      PLAT1=ASIN(ZT/RS)*RD
   43 T1=T2
      CALL TCONIC(U,E,A,F+1.,T2)
      DT=T2-T1
      IF(DT.LT.0.) DT=PERIOD+DT
      CALL UPDATE(DT,1.)
      TSTEP=TSTEP+DT $ FSTEP=FSTEP+1.
      CALL RAY(0.,0.,XT,YT,ZT,KK)
      IF(KK.EQ.0) GO TO 799
      PLAT2=ASIN(ZT/RS)*RD
      PLWRTF=(PLAT2-PLAT1)/1.
      DF=(PLAT-PLAT1)/PLWRTF-1.
      T1=T2
      CALL TCONIC(U,E,A,F+DF,T2)
      DT=T2-T1
      IF(DT.LT.0..AND.DF.GT.0.) DT=PERIOD+DT
      IF(DT.GT.0..AND.DF.LT.0.) DT=DT-PERIOD
      CALL UPDATE(DT,DF)
      TSTEP=TSTEP+DT $ FSTEP=FSTEP+DF
      CALL RAY(0.,0.,XT,YT,ZT,KK)
      IF(KK.EQ.0) GO TO 799
      PLAT1=ASIN(ZT/RS)*RD
      IF(ABS(PLAT1-PLAT).LT..001) 45,44
   44 NCNT=NCNT+1
      IF(NCNT.EQ.10) 799,43
   45 CALL UPDATE(-TSTEP,-FSTEP)
      GO TO 50
   46 CONTINUE
```

LOGIC

```
C
C         DUMMY LOGIC
C
      FSTEP=100.
      GO TO 37
C
   52 SXI=SIN(XI*DR)
      SC1=SIN(LAT1*DR)/SXI
      SC3=SIN(LAT2*DR)/SXI
      CO=ANGLE(W+F)
      IF(ABS(SC1).GE.1.) 55,56
   55 C1=CO+359.99 $ C2=C1  $ GO TO 57
   56 CONTINUE
      C1=ASIN(SC1)*RD
      C2=SIGN(180.,C1)-C1
   57 IF(ABS(SC3).GE.1.) 58,59
   58 C3=CO+359.99 $ C4=C3 $ GO TO 60
   59 CONTINUE
      C3=ASIN(SC3)*RD
      C4=SIGN(180.,C3)-C3
   60 CONTINUE
      CMIN=AMIN1(C1,C2,C3,C4)
      IF(CMIN.GT.999.) GO TO 799
      DELC=AMIN1(ANGLE(C1-CO),ANGLE(C2-CO),ANGLE(C3-CO),ANGLE(C4-CO))
      IF(ABS(DELC-ANGLE(C1-CO)).LT..001) GO TO 53
      IF(ABS(DELC-ANGLE(C2-CO)).LT..001) GO TO 53
      PLAT=LAT2
      LATLAST=LAT1
      GO TO 54
   53 PLAT=LAT1
      LATLAST=LAT2
   54 GO TO 41
C
   50 GO TO (51,51,9),IPG
C
   51 FLAGP=1
      NUMP=1
      JJ=2
      GO TO 800
C
  799 JJ=3
  800 RETURN
      END
```

APPENDIX B

```
      SUBROUTINE GPRINT
C
C     THIS SUBROUTINE CALCULATES AND WRITES GROUNDTRACK DATA
C
      DIMENSION DATE(6),OTME(4),FOOTL(3,4)
      REAL JDO,JD,LAT3,LON2
      REAL LAT1,LAT2,LAMBDA,LATLAST
      INTEGER CSTOP,CNUM,OLAST
      COMMON/BLOCK1/JDO,JD,DATE,CNUM,HA,OLAST,DELT,DELF
      COMMON/BLOCK2/A,E,XI,W,O,F,PERIOD,ANGRAT,FO,FTOTAL,TTOTAL
      COMMON/BLOCK3/RD,DR,U,RS,PI,HAREF,HADOT,JDREF
      COMMON/BLOCK4/SX,SY,SZ,EX,EY,EZ,CX,CY,CZ
      COMMON/BLOCK5/IPG,IPUT,IDATE,ISTEP,IOCC,IPHOTO,IMANU,IB,IOLAP,IFRA
     *ME,ISINGLE
      COMMON/BLOCK6/TSTOP,FSTOP,CSTOP
      COMMON/BLOCK7/HAO,HPO,AO,EO,XIO,WO,OO,VINF,SDEC,SRA,BETA
      COMMON/BLOCK8/XJ20,RES,PHI1,PHI2,CVLAP,CANGF,CANGS,FPHOTO1,FPHOTO2
     *,FOOTL,LAMBDA,LAT1,LAT2,PLAT
      ANGLE(X)=AMOD(X,360.)+180.-SIGN(180.,X)
      CALL CONCAR(A,E,XI,W,O,F,X1,Y1,Z1,DX1,DY1,DZ1,U)
      X2=X1*COS(HA*DR)+Y1*SIN(HA*DR)
      Y2=-X1*SIN(HA*DR)+Y1*COS(HA*DR)
      Z2=Z1
      DX2=DX1*COS(HA*DR)+DY1*SIN(HA*DR)
      DY2=-DX1*SIN(HA*DR)+DY1*COS(HA*DR)
      DZ2=DZ1
C
      CALL DOT(X1,Y1,Z1,SX,SY,SZ,PHI)
      CALL CARCON(X2,Y2,Z2,DX2,DY2,DZ2,A2,E2,XI2,W2,O2,F2,U)
C
      CALL CARSPH(X2,Y2,Z2,DX2,DY2,DZ2,R2,LON2,LAT3,V2,GAM2,SIG2)
      LON2=ANGLE(LON2)
      H2=R2-RS
C
      WM=HADOT*DR/86400.
      VHI=V2*COS(GAM2*DR)
      VHIE=VHI*SIN(SIG2*DR)
      VHIN=VHI*COS(SIG2*DR)
      VME=RS*COS(LAT3*DR)*WM
      VTE=VHIE-VME
      SIG2R=ATAN2(VTE,VHIN)*RD
      VHR=SQRT(VHIN**2+VTE**2)
```

GPRINT

```
      HVOH=VHR/H2
      DELJD=TTOTAL/86400.
      DAYS=DELJD-AMOD(DELJD,1.)
      DELHRS=(DELJD-DAYS)*24.
      HRS=DELHRS-AMOD(DELHRS,1.)
      DELMIN=(DELHRS-HRS)*60.
      MIN=DELMIN-AMOD(DELMIN,1.)
      SEC=(DELMIN-MIN)*60.
      OTME(1)=DAYS
      OTME(2)=HRS
      OTME(3)=MIN
      OTME(4)=SEC
      S=TTOTAL
C
C
      WRITE(6,101)JD,DATE,OTME,S,ONUM
      WRITE(6,102)LAT3,LON2,H2,V2,GAM2,PHI,HVOH,HA,W,O,F
  101 FORMAT(1H0/* JULIAN DATE*,F18.8,2X,*CALENDAR DATE*,F4.0,4F3.0,F7.2
     1,2X,*ORBIT TIME*,3F3.0,F6.2,2X,*(SEC)*,G12.6,*ORBIT NO*,I2)
  102 FORMAT(1H0,*LAT  =*,E16.8,2X,*LONG =*,E16.8,2X,*ALT  =*,E16.8,2X,
     1*VELI =*,E16.8,2X,*FPAI =*,E16.8,/1X,*PHI  =*,E16.8,2X,*VOHR =*,
     2E16.8,2X,*HA   =*,E16.8,2X,*W    =*,E16.8,2X,*O    =*,E16.8,/1X,*T
     3.A. =*,E16.8)
      RETURN
      END
```

```
      SUBROUTINE CPRINT(K)
C
C     THIS SUBROUTINE WRITES THE PROGRAM INPUTS
C
C     K=0  INITAL PRINTOUT ( START OF CASE )
C     K=1  PERIAPSIS PRINTOUT
C
      COMMON/BLOCK1/JDC,JD,DATE,CNUM,HA,OLAST,DELT,DELF
      COMMON/BLOCK2/A,E,XI,W,O,F,PERIOD,ANGRAT,FO,FTOTAL,TTOTAL
      COMMON/BLOCK3/RD,DR,U,RS,PI,HAREF,HADOT,JDREF
      COMMON/BLOCK4/SX,SY,SZ,EX,EY,EZ,CX,CY,CZ
      COMMON/BLOCK5/IPG,IPUT,IDATE,ISTEP,IOCC,IPHOTO,IMANU,IB,IOLAP,IFRA
     *ME,ISINGLE
      COMMON/BLOCK6/TSTOP,FSTOP,CSTOP
      COMMON/BLOCK7/HAC,HPO,AO,EO,XIC,WC,OO,VINF,SDEC,SRA,BETA
      COMMON/BLOCK8/XJ20,RES,PHI1,PHI2,OVLAP,CANGF,CANGS,FPHOTO1,FPHOTO2
     *,FOOTL,LAMBDA,LAT1,LAT2,PLAT
      DIMENSION DATE(6),FOOTL(3,4)
      REAL LAT1,LAT2,LAMBDA,LATLAST
      INTEGER CSTOP
C
      KK=K+1
      GO TO (1,2),KK
    1 CONTINUE
      WRITE(6,1001)AO,BETA,CANGF,CANGS,DATE(1),DATE(2),DATE(3),DATE(4),D
     1ATE(5),DATE(6),DELF,DELT,EC,FC,FPHOTO1,FPHOTO2,FSTOP,HAO,HPO,IB,ID
     2ATE,IFRAME,IMANU,IOCC,IOLAP,IPG,IPHOTO,IPUT,ISINGLE,ISTEP,JD,
     3LAMBDA,LAT1,LAT2,OO,CSTOP,OVLAP,PHI1,PHI2,PLAT,RES,RS,SDEC,SRA,
     4TSTOP,U,VINF,WC,XIC,XJ20
 1001 FORMAT(*1$CASE*//* AO       =*,E16.8,*      BETA     =*,E16.8,*
     1CANGF    =*,E16.8,*       CANGS    =*,E16.8/* DATE(1) =*,E16.8,*
     2DATE(2) =*,E16.8,*        DATE(3) =*,E16.8,*       DATE(4) =*,E16.8/*
     3 DATE(5) =*,E16.8,*       DATE(6) =*,E16.8,*       DELF     =*,E16.8,*
     4       DELT     =*,E16.8/* EC       =*,E16.8,*      FC       =*,E16.8,*
     5       FPHOTO1 =*,E16.8,*       FPHOTO2 =*,E16.8/* FSTOP    =*,E16.8,
     6*      HAO      =*,E16.8,*       HPO      =*,E16.8,*       IB       =*,I
     716/* IDATE    =*,I16,*      IFRAME   =*,I16,*       IMANU    =*,I16,*
     8    IOCC     =*,I16/* IOLAP    =*,I16,*       IPG      =*,I16,*       IP
     9HOTO   =*,I16,*       IPUT     =*,I16/* ISINGLE =*,I16,*       ISTEP
     *=*,I16,*                  *,16X,*      JD       =*,I16/* LAMBDA   =*,E1
     *6.8,*       LAT1     =*,E16.8,*       LAT2     =*,E16.8,*       OO
     *=*,E16.8/* CSTOP    =*,I16,*      OVLAP    =*,E16.8,*       PHI1     =*
```

OPRINT

```
     *,E16.8,*       PHI2     =*,E16.8/* PLAT     =*,E16.8,*      RES       =*
     *,E16.8,*       RS       =*,E16.8,*      SDEC      =*,E16.8/* SRA       =
     **,E16.8,*      TSTOP    =*,E16.8,*      U         =*,E16.8,*         VINF
     *     =*,E16.8/* WO       =*,E16.8,*      XIO       =*,E16.8,*         XJ2C
     *     =*,E16.8)
      IF(IPG.EQ.1)WRITE(6,106)
  106 FORMAT(*0GROUND TRACK OPTION*)
      IF(IPG.EQ.2)WRITE(6,107)
  107 FORMAT(*0PHOTO OPTION*)
      IF(IPG.EQ.3)WRITE(6,108)
  108 FORMAT(*0GROUND TRACK AND PHOTO OPTION*)
      IF(IPUT.EQ.1)WRITE(6,109)AO,EO
  109 FORMAT(* INPUT  *,*  AO =*,E16.8,*  EO =*,E16.8)
      IF(IPUT.EQ.2)WRITE(6,110)HPC,HAC
  110 FORMAT(* INPUT  *,*  HPC =*,E16.8,*  HAC =*,E16.8)
      IF(IPUT.EQ.3)WRITE(6,111)
  111 FORMAT(* S  VECTOR  INPUT*)
      IF(IDATE.EQ.1)WRITE(6,112)
  112 FORMAT(* CALENDAR  DATE  INPUT*)
      IF(IDATE.EQ.2)WRITE(6,113)
  113 FORMAT(* JULIAN  DATE  INPUT*)
      IF(ISTEP.EQ.1)WRITE(6,114)DELT
  114 FORMAT(* TIME  STEP  OF  *,E16.8,*  EMPLOYED*)
      IF(ISTEP.EQ.2)WRITE(6,115)DELF
  115 FORMAT(* TRUE  ANOMALY  STEP  OF  *,E16.8,*  EMPLOYED*)
      IF(IOCC.EQ.0)WRITE(6,116)
  116 FORMAT(* NO  OCCULATION  DATA*)
      IF(IOCC.EQ.1)WRITE(6,117)
  117 FORMAT(* OCCULATION  DATA  GIVEN*)
      IF(IPHOTO.EQ.0)WRITE(6,118)
  118 FORMAT(* VERTICAL  PHOTOGRAPHY*)
      IF(IPHOTO.EQ.1)WRITE(6,119)
  119 FORMAT(* NON  VERTICAL  PHOTOGRAPHY*)
      IF(IMANU.EQ.0)WRITE(6,120)
  120 FORMAT(* NO  MANEUVERS  MADE*)
      IF(IMANU.EQ.1)WRITE(6,121)
  121 FORMAT(* MANEUVERS  COMPUTED*)
      IF(IB.EQ.0)WRITE(6,122)
  122 FORMAT(* BMATRIX  NOT  COMPUTED*)
      IF(IB.EQ.1)WRITE(6,123)
  123 FORMAT(* BMATRIX  COMPUTED*)
      IF(ICLAP.EQ.0)WRITE(6,124)
  124 FORMAT(* PICTURES  TAKEN  ON  STEP*)
```

OPRINT

```
      IF(IOLAP.EQ.1)WRITE(6,125)CVLAP
  125 FORMAT(* PICTURES  TAKEN  CN  *,E16.8,*  OVERLAP*)
      IF(IFRAME.EQ.1)WRITE(6,126)
  126 FORMAT(* SINGLE  PICTURE  TAKEN*)
      IF(IFRAME.EQ.2)WRITE(6,127)PHI1,PHI2
  127 FORMAT(* PICTURES  TAKEN  BETWEEN  PHI1  OF *,E16.8,*  AND  PHI2
     1OF  *,E16.8)
      IF(IFRAME.EQ.3)WRITE(6,128)FPHCTC1,FPHOTO2
  128 FORMAT(* PICTURES  TAKEN  BETWEEN  FPHOTO1  OF *,F7.2,*  AND  FPHC
     1TO2  OF *,F7.2)
      IF(IFRAME.EQ.4)WRITE(6,129)LAT1,LAT2
  129 FORMAT(* PICTURES  TAKEN  BETWEEN  LAT1  OF  *,F7.2,*  AND  LAT2
     1OF  *,F7.2)
      TOSTCP=360.-FO+360.*(OSTOP-1)
      TFSTOP=FSTOP
      B=AMOD(TSTOP,PERICD)
      CALL TCONIC(U,E,A,FO,TFO)
      XM=(TFO+B)*ANGRAT
      IF(XM.GT.2.*PI) 3,4
    3 B=B-PERIOD
      XM=XM-2.*PI
    4 CONTINUE
      CALL TINVS(XM,E,EC,FB)
      FOB=FB*RD-FC
      IF(FOB.LT..001)FOB=360.+FOB
      TTSTOP=360.*(TSTCP-B)/PERICD+FOB
      IF(TOSTOP.LT.TFSTOP.AND.TCSTOP.LT.TTSTOP)WRITE(6,2000)OSTOP
 2000 FORMAT(* PROGRAM  WILL  STCP  CN  OSTOP = *,I5)
      IF(TFSTOP.LT.TTSTOP.AND.TFSTOP.LT.TCSTOP)WRITE(6,2001)FSTOP
 2001 FORMAT(* PROGRAM  WILL  STCP  CN  FSTOP = *,E16.8,*  DEGREES*)
      IF(TTSTOP.LT.TFSTOP.AND.TTSTOP.LT.TOSTOP)WRITE(6,2002)TSTOP
 2002 FORMAT(* PROGRAM  WILL  STCP  CN  TSTOP = *,E16.8,*  SEC*)
C
    2 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE OVERLAP(TSTEP)
C
C     THIS SUBROUTINE CALCULATES THE TIME INCREMENT TO THE NEXT CAMERA
C     FOOTPRINT FOR A GIVEN OVERLAP RATIO
C
C     TSTEP - TIME TO NEXT CAMERA FOOTPRINT
C
      COMMON/BLOCK1/JDO,JD,DATE,CNUM,HA,OLAST,DELT,DELF
      COMMON/BLOCK2/A,E,XI,W,O,F,PERIOD,ANGRAT,FC,FTOTAL,TTOTAL
      COMMON/BLOCK3/RD,DR,U,RS,PI,HAREF,HADOT,JDREF
      COMMON/BLOCK5/IPG,IPUT,IDATE,ISTEP,ICCC,IPHOTO,IMANU,IB,IOLAP, IFRA
     *ME,ISINGLE
      COMMON/BLOCK8/XJ20,RES,PHI1,PHI2,OVLAP,CANGF,CANGS,FPHOTO1,FPHOTO2
     *,FOOTL,LAMBDA,LAT1,LAT2,PLAT
      REAL LAT1,LAT2,LAMBDA,LATLAST
C
      DIMENSION FOOTO(3,4),FOOTF(3,4),K(4),KF(4),DATE(6),FOOTL(3,4)
C
      ANGLE(X)=AMOD(X,360.)+180.-SIGN(180.,X)
C
      CALL RAY(-CANGF,-CANGS,FOOTO(1,1),FOOTO(2,1),FOOTO(3,1),K(1))
      CALL RAY(-CANGF, CANGS,FOOTO(1,2),FOOTO(2,2),FOOTO(3,2),K(2))
      CALL RAY( CANGF, CANGS,FOOTO(1,3),FOOTO(2,3),FOOTO(3,3),K(3))
      CALL RAY( CANGF,-CANGS,FOOTO(1,4),FOOTO(2,4),FOOTO(3,4),K(4))
C
      IF(K(1)+K(2)+K(3)+K(4).EQ.4) GO TO 7
C
      DF=0 $ TSTEP=0 $ CALL TCONIC(U,E,A,F,T1)
    8 TSTEP=TSTEP+100.
      T2=T1+TSTEP
      XM=ANGRAT*T2
      CALL TINVS(XM,E,EC2,F2)
      FSTEP=ANGLE(ANGLE(F2*RD)-F)
      DF=DF+FSTEP
      CALL UPDATE(100.,FSTEP)
      CALL RAY(-CANGF,-CANGS,X,Y,Z,K(1))
      CALL RAY(-CANGF, CANGS,X,Y,Z,K(2))
      CALL RAY( CANGF, CANGS,X,Y,Z,K(3))
      CALL RAY( CANGF,-CANGS,X,Y,Z,K(4))
      IF(K(1)+K(2)+K(3)+K(4).NE.4) GO TO 8
      CALL UPDATE(-TSTEP,-DF)
      GO TO 800
```

APPENDIX B

OVERLAP

```
    7 CONTINUE
C
      SHA=SIN(HA*DR) $ CHA=COS(HA*DR)
C
      DO 1 I=1,4
      XTEMP=FOOTO(1,I) $ YTEMP=FOOTO(2,I)
      FOOTO(1,I)=XTEMP*CHA+YTEMP*SHA
    1 FOOTO(2,I)=-XTEMP*SHA+YTEMP*CHA
C
C
      DT=0. $ OLAP1=1.
      DF=0. $ FSTEP=1.
C
      CALL TCONIC(U,E,A,F,T1)
C
    2 CONTINUE
C
      CALL TCONIC(U,E,A,F+FSTEP,T2)
      TSTEP=T2-T1
      IF(TSTEP.LT.0..AND.FSTEP.GT.0.) TSTEP=PERIOD+TSTEP
      IF(TSTEP.GT.0..AND.FSTEP.LT.0.) TSTEP=TSTEP-PERIOD
      DF=DF+FSTEP
      DT=DT+TSTEP
      CALL UPDATE(TSTEP,FSTEP)
      CALL RAY(-CANGF,-CANGS,FOOTF(1,1),FOOTF(2,1),FOOTF(3,1),KF(1))
      CALL RAY(-CANGF, CANGS,FOOTF(1,2),FOOTF(2,2),FOOTF(3,2),KF(2))
      CALL RAY( CANGF, CANGS,FOOTF(1,3),FOOTF(2,3),FOOTF(3,3),KF(3))
      CALL RAY( CANGF,-CANGS,FOOTF(1,4),FOOTF(2,4),FOOTF(3,4),KF(4))
C
      SHA=SIN(HA*DR) $ CHA=COS(HA*DR)
C
      DO 3 I=1,4
      XTEMP=FOOTF(1,I) $ YTEMP=FOOTF(2,I)
      FOOTF(1,I)=XTEMP*CHA+YTEMP*SHA
    3 FOOTF(2,I)=-XTEMP*SHA+YTEMP*CHA
C
      CALL PRATIO(FOOTO,FOOTF,RATIO)
      IF(RATIO.GT..99999)GO TO 6
C
      IF(ABS(RATIO).GT..000001) GO TO 5
      CALL UPDATE(-TSTEP,-FSTEP)
      DF=DF-FSTEP $ DT=DT-TSTEP
      FSTEP=FSTEP/2.
```

OVERLAP

```
      GO TO 2
    5 CONTINUE
      IF(ABS(RATIO-OVLAP).LT..C001) GO TO 4
      POWRTF=(RATIO-OLAP1)/FSTEP
      FSTEP=-(RATIO-OVLAP)/POWRTF
      IF(POWRTF.GT.C.0) FSTEP=5.
      IF(ABS(FSTEP).GT.5.) FSTEP=SIGN(5.,FSTEP)
      CLAP1=RATIO
      T1=T2
      GO TO 2
    6 FSTEP=5.
      T1=T2
      GO TO 2
C
    4 CONTINUE
      TSTEP=DT
      CALL UPDATE(-DT,-DF)
C
  800 RETURN
      END
```

APPENDIX B

```
      SUBROUTINE PPRINT
C
C     THUS SUBRCUTINE CALCULATES AND WRITES PHOTOGRAPHY DATA
C
      COMMCN/BLCCK1/JDC,JD,CATE,CNUM,HA,OLAST,DELT,DELF
      COMMON/BLCCK2/A,E,XI,W,O,F,PERICC,ANGRAT,FO,FTOTAL,TTCTAL
      COMMCN/BLCCK3/RD,DR,U,RS,PI,HAREF,HADOT,JDREF
      COMMON/BLCCK4/SX,SY,SZ,EX,EY,EZ,CX,CY,CZ
      COMMON/BLOCK5/IPG,IPUT,ICATE,ISTEP,IOCC,IPHOTO,IMANU,IB,IOLAP,IFRA
     *ME,ISINGLE
      COMMCN/BLCCK6/TSTOP,FSTCP,CSTOP
      COMMCN/BLCCK7/HAC,HPO,AC,EC,XIO,WC,OO,VINF,SDEC,SRA,BETA
      COMMON/BLOCK8/XJ20,RES,PHI1,PHI2,CVLAP,CANGF,CANGS,FPHOTO1,FPHCTC2
     *,FOOTL,LAMECA,LAT1,LAT2,PLAT
      COMMON/BLOCK9/NUMP
      REAL LAT1,LAT2,LAMBDA,LATLAST
      DIMENSION DATE(6),FOOTL(3,4)
      DIMENSICN FOCT(3,4),KK(8)
      ANGLE(X)=AMOD(X,360.)+180.-SIGN(180.,X)
C
      CALL GPRINT
C
      CALL RAY (-CANGF,-CANGS,FCCT(1,1),FOOT(2,1),FOOT(3,1),KK(1))
      CALL RAY (-CANGF, CANGS,FCCT(1,2),FOOT(2,2),FOCT(3,2),KK(3))
      CALL RAY ( CANGF, CANGS,FCCT(1,3),FOOT(2,3),FOOT(3,3),KK(5))
      CALL RAY ( CANGF,-CANGS,FOOT(1,4),FOOT(2,4),FCOT(3,4),KK(7))
C
      IF(KK(1)+KK(3)+KK(5)+KK(7).NE.4) GO TC 300
C
      SHA=SIN(HA*DR) $ CHA=COS(HA*DR)
C
      DO 1 I=1,4
      XTEMP=FOOT(1,I) $ YTEMP=FCCT(2,I)
      FOOT(1,I)= XTEMP*CHA+YTEMP*SHA
    1 FOOT(2,I)=-XTEMP*SHA+YTEMP*CHA
C
      CALL PRATIC(FOOTL,FOOT,RATIO)
C
      DO 201 I=1,3
      DO 201 J=1,4
  201 FOOTL(I,J)=FCCT(I,J)
C
```

PPRINT

```
      WRITE(6,101) RATIO,NUMP
  101 FORMAT(* PHOTO OVERLAP = *F7.3,10X,*FOOTPRINT NUMBER = *I3/12X,*LA
     *TITUDE*,11X,*LONGITUDE*,12X,*SUN ANGLE*,10X,*SLANT RANGE*,7X,*STAT
     *IC RESOLUTION*)
C
      CALL CONCAR(A,E,XI,W,O,F,X,Y,Z,DX,DY,DZ,U)
C
      XTEMP=X $ YTEMP=Y
      X=XTEMP*CHA+YTEMP*SHA
      Y=-XTEMP*SHA+YTEMP*CHA
C
      XSUN=SX*CHA+SY*SHA
      YSUN=-SX*SHA+SY*CHA
      ZSUN=SZ
C
      N=0
    9 N=N+1
      GO TO (10,11,12,11,13,11,14,11,15,200),N
C
   10 CALL RAY(0.,0.,XT,YT,ZT,K)
      GO TO 16
C
   11 J=N/2
      XT=FOOT(1,J)
      YT=FOOT(2,J)
      ZT=FOOT(3,J)
      GO TO 17
C
   12 CALL RAY(-CANGF,0.,XT,YT,ZT,KK(N-1))
      GO TO 16
   13 CALL RAY(0.,CANGS,XT,YT,ZT,KK(N-1))
      GO TO 16
   14 CALL RAY(CANGF,0.,XT,YT,ZT,KK(N-1))
      GO TO 16
   15 CALL RAY(0.,-CANGS,XT,YT,ZT,KK(N-1))
      GO TO 16
C
   16 XTEMP=XT $ YTEMP=YT
      XT=XTEMP*CHA+YTEMP*SHA
      YT=-XTEMP*SHA+YTEMP*CHA
C
   17 CALL LATLNG(XT,YT,ZT,XLAT,XLONG)
      XLONG=ANGLE(XLONG)
```

APPENDIX B

PPRINT

```
      CALL DOT(XT,YT,ZT,XSUN,YSUN,ZSUN,PHI)
      SR=SQRT((X-XT)**2+(Y-YT)**2+(Z-ZT)**2)
      SRES=SR*RES
C
      GO TO (18,19,19,19,19,19,19,19,19),N
C
   18 WRITE(6,102) XLAT,XLONG,PHI,SR,SRES
  102 FORMAT(* CENTER*,6X,F6.2,13X,F7.2,13X,F7.2,13X,F9.3,13X,F9.2)
      GO TO 9
C
   19 I=N-1
      WRITE(6,103) I,XLAT,XLONG,PHI,SR,SRES
  103 FORMAT(* POINT*,I2,5X,F6.2,13X,F7.2,13X,F7.2,13X,F9.3,13X,F9.2)
      GO TO 9
C
  300 WRITE(6,104)
  104 FORMAT(*0PHOTO FOOTPRINT OFF PLANET *)
  200 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE RAY(FANGLE,SANGLE,XT,YT,ZT,KK)
C
C     THIS SUBROUTINE CALCULATES THE POINT WHERE THE CAMERA VECTOR PIERCES
C     THE SURFACE OF THE PLANET
C
C     FANGLE - FORWARD CAMERA HALF ANGLE
C     SANGLE - SIDE CAMERA HALF ANGLE
C     XT,YT,ZT - POINT WHERE THE CAMERA VECTOR PIERCES THE SURFACE
C     KK - 0 IMPLIES DOES NOT PIERCE THE SURFACE-1 IMPLIES DOES PIERCE
C     THE SURFACE
C
      COMMON/BLOCK2/A,E,XI,W,O,F,PERIOD,ANGRAT,FC,FTOTAL,TTOTAL
      COMMON/BLOCK3/RD,DR,U,RS,PI,HAREF,HADOT,JDREF
      COMMON/BLOCK5/IPG,IPUT,IDATE,ISTEP,ICCC,IPHOTO,IMANU,IB,IOLAP,IFRA
     *ME,ISINGLE
      COMMON/BLOCK8/XJ20,RES,PHI1,PHI2,CVLAP,CANGF,CANGS,FPHOTO1,FPHOTO2
     *,FOOTL,LAMBDA,LAT1,LAT2,PLAT
      REAL LAT1,LAT2,LAMBDA,LATLAST
      REAL JC,JCO,JDREF
      DIMENSION FOOTL(3,4)
      KK=1
      J=IPHOTO+1
      GO TO (10,30),J
   10 CONTINUE
      SF=SIN(FANGLE*DR) $ CF=COS(FANGLE*DR)
      SS=SIN(SANGLE*DR) $ CS=COS(SANGLE*DR)
   11 CALL CONCAR(A,E,XI,W,O,F,X,Y,Z,DX,DY,DZ,U)
      R=SQRT(X*X+Y*Y+Z*Z)
      H1=-X/R $ H2=-Y/R $ H3=-Z/R
      P1=H2*DZ-H3*DY $ P2=H3*DX-H1*DZ $ P3=H1*DY-H2*DX
      P=SQRT(P1*P1+P2*P2+P3*P3)
      P1=P1/P $ P2=P2/P $ P3=P3/P
      Q1=P2*H3-P3*H2 $ Q2=P3*H1-P1*H3 $ Q3=P1*H2-P2*H1
      RH=CS*CF $ RQ=CS*SF $ RP=SS
      R1=RH*H1+RQ*Q1+RP*P1
      R2=RH*H2+RQ*Q2+RP*P2
      R3=RH*H3+RQ*Q3+RP*P3
      SINBC=RS/R $ COSBC=SQRT(1.-SINBC**2)
      IF(RH.GT.COSBC) GO TO 20
      KK=0
      GO TO 800
   20 SR=R*RH-SQRT((R*RH)**2-R**2+RS**2)
      XT=X+R1*SR $ YT=Y+R2*SR $ ZT=Z+R3*SR
      GO TO 800
   30 CONTINUE
      SF=SIN(FANGLE*DR) $ CF=COS(FANGLE*DR)
      SS=SIN((SANGLE+LAMBDA)*DR) $ CS=COS((SANGLE+LAMBDA)*DR)
      GO TO 11
  800 RETURN
      END
```

APPENDIX B

```
      SUBROUTINE STDCASE
C
C     THIS SUBROUTINE DEFINES PROGRAM CONSTANTS AND STANDARD INPUT DATA
C
      COMMON/BLOCK1/JDO,JD,DATE,ONUM,HA,OLAST,DELT,DELF
      COMMON/BLOCK2/A,E,XI,W,O,F,PERIOD,ANGRAT,FO,FTOTAL,TTOTAL
      COMMON/BLOCK3/RD,DR,U,RS,PI,HAREF,HADOT,JDREF
      COMMON/BLOCK4/SX,SY,SZ,EX,EY,EZ,CX,CY,CZ
      COMMON/BLOCK5/IPG,IPUT,IDATE,ISTEP,IOCC,IPHOTO,IMANU,IB,IOLAP,IFRA
     *ME,ISINGLE
      COMMON/BLOCK6/TSTOP,FSTOP,OSTOP
      COMMON/BLOCK7/HAO,HPO,AO,EO,XIO,WO,OO,VINF,SDEC,SRA,BETA
      COMMON/BLOCK8/XJ20,RES,PHI1,PHI2,OVLAP,CANGF,CANGS,FPHOTO1,FPHOTO2
     *,FOOTL,LAMBDA,LAT1,LAT2,PLAT
C
      REAL JD,JDO,JDREF
      REAL LAT1,LAT2,LAMBDA,LATLAST
      INTEGER OSTOP,ONUM,OLAST
      DIMENSION DATE(6),FOOTL(3,4)
C
      DATA RD,DR,PI/57.2957795130823,.0174532925199433,3.14159265358979/
C
      FTOTAL=0.
      TTOTAL=0.
      ONUM=1
      OLAST=1
      IOCC=0
      IMANU=0
      IB=0
      DO 1 I=1,2
      DO 1 J=1,4
    1 FOOTL(I,J)=0.
      DO 2 J=1,4
    2 FOOTL(3,J)=1.
C
      TSTOP=1.E+10
      FSTOP=1.E+10
      OSTOP=10000
      U=    42829.5
      RS=    3386.
      HAREF=149.475
      HADOT=350.891962
      JDREF=2418322.0
C
      XJ20=0.00195
      RES=10.
C
      RETURN
      END
```

```
      SUBROUTINE UPDATE(TSTEP,FSTEP)
C
C     THIS SUBROUTINE UPDATES ALL TIME DEPENDENT PARAMETERS SUCH AS
C     TRUE ANOMALY ,HOUR ANGLE , ETC.
C
C     TSTEP - INCREASE TIME BY TSTEP
C     FSTEP - INCREASE TRUE ANOMALY BY FSTEP
C
      ANGLE(X)=AMOD(X,360.)+180.-SIGN(180.,X)
      COMMON/BLOCK1/JDO,JD,DATE,CNUM,HA,OLAST,DELT,DELF
      COMMON/BLOCK2/A,E,XI,W,O,F,PERIOD,ANGRAT,FO,FTOTAL,TTOTAL
      COMMON/BLOCK3/RD,DR,U,RS,PI,HAREF,HADOT,JDREF
      COMMON/BLOCK4/SX,SY,SZ,EX,EY,EZ,CX,CY,CZ
      COMMON/BLOCK5/IPG,IPUT,IDATE,ISTEP,IOCC,IPHOTO,IMANU,IB,IOLAP,IFRA
     *ME,ISINGLE
      COMMON/BLOCK6/TSTOP,FSTOP,CSTOP
      COMMON/BLOCK7/HAO,HPO,AO,EO,XIO,WO,OO,VINF,SDEC,SRA,BETA
      COMMON/BLOCK8/XJ20,RES,PHI1,PHI2,OVLAP,CANGF,CANGS,FPHOTO1,FPHOTO2
     *,FOOTL,LAMBDA,LAT1,LAT2,PLAT
      REAL JD,JDO,JDREF
      REAL LAT1,LAT2,LAMBDA,LATLAST
      INTEGER CSTOP,CNUM,OLAST
      DIMENSION DATE(6),FOOTL(3,4)
      JD=JD+TSTEP/3600./24
      WDI=FLOAT(IFIX(JD))
      FDI=JD-WDI
      CALL JULCAL(DATE,WDI,FDI,0)
      F=ANGLE(F+FSTEP)
      FTOTAL=FTOTAL+FSTEP
      TTOTAL=TTOTAL+TSTEP
      OLAST=CNUM
      ONUM=(FTOTAL+FO)/360.+1.0
      IF(OLAST.GT.ONUM)OLAST=ONUM
      HA=HAREF+HADOT*(JD-JDREF)
      HA=ANGLE(HA)
      CALL WELLS(RS,U,XJ20,A,E,XI,W,O,TSTEP)
      CALL VECTOR(JD,DECS,RAS,DECE,RAE,DECC,RAC,SX,SY,SZ,EX,EY,EZ,CX,CY,
     1CZ,4)
      RETURN
      END
```

APPENDIX B

```
      SUBROUTINE CALJUL(WJD,FJD,WND,FD,X)
C
C     THIS SUBROUTINE CONVERTS A GIVEN CALENDAR DATE TO THE CORRESPONDING
C     JULIAN DATE AND ALSO COMPUTES THE NUMBER OF DAYS WHICH HAVE ELAPSED
C     SINCE JANUARY 1 AT 0000 HOURS,1950.
C
C     WJD - WHOLE NUMBER PART OF JULIAN DATE
C     FJD - FRACTIONAL PART OF JULIAN DATE
C     WND - WHOLE NUMBER OF DAYS SINCE JANUARY 1.0,1950
C     FDA - FRACTIONAL PART OF WND
C     X(1-6) - CALENDAR DATE (YEAR,MONTH,DAY,HOUR,MINUTE,SECOND)
C
      DIMENSION X(6),A(12)
      D50=2433282.
      YD=X(1)-48.
      YL=YD/4.
      KYL=YL
      CK=KYL
      IF(YL-CK)1,1,3
    1 IF(X(2)-2.)4,4,3
    3 DS=CK
      GO TO 5
    4 DS=CK-1.
    5 DS=DS+365.*(YD-2.)
      DO 6 I=1,12
    6 A(I)=1.0
      K=X(2)
      DO 7 I=K,12
    7 A(I)=0.0
      DS=DS+31.*(A(1)+A(3)+A(5)+A(7)+A(8)+A(10)+A(12))
     1+30.*(A(4)+A(6)+A(9)+A(11))+28.*A(2)
      DS=DS+X(3)-1.
      WND=DS
      FD=X(4)/24.+X(5)/1440.+X(6)/86400.
      IF(FD-.4999999)9,8,8
    8 FJD=FD-.5
      WJD=1.
      GO TO 10
    9 FJD=FD+.5
      WJD=0.
   10 WJD=D50+WJD+WND
      RETURN
      END
```

```
      SUBROUTINE CARCON(X,Y,Z,DX,DY,DZ,A,E,XI,W,O,F,U)
C
C     THIS SUBRCUTINE CCNVERTS CARTESIAN COORDINATES TO CONIC ELEMENTS
C
C     X,Y,Z - CCMPCNENTS OF PCSITION VECTOR
C     DX,DY,DZ - COMPONENTS OF VELOCITY VECTCR
C     A,E,XI - SEMI-MAJOR AXIS, ECCENTRICITY, INCLINATION
C     W,O,F - ARGUMENT OF PERIAPSIS, LCNGITUDE OF ASCENDING NODE, TRUE
C             ANOMALY
C     U - GRAVITIONAL CCNSTANT
C
      DATA RD/57.2957795130823/
      ANGLE(X)=AMOD(X,360.)+180.-SIGN(180.,X)
      C1=Y*DZ-Z*DY
      C2=Z*DX-X*DZ
      C3=X*DY-Y*DX
      H=SQRT(C1*C1+C2*C2+C3*C3)
      CXIR=C3/H
      XIR=ATAN2(SQRT(1.-CXIR**2),CXIR)
      SXIR=SIN(XIR)
      SO=0.
      CO=1.
      IF(SXIR.EQ.0.)GO TO 5
      SO=C1/(H*SXIR)
      CO=-C2/(H*SXIR)
    5 R=SQRT(X*X+Y*Y+Z*Z)
      V=SQRT(DX*DX+DY*DY+DZ*DZ)
      A=R*U/(2.*U-R*V*V)
      E=SQRT(1.-H*H/(U*A))
      P=A*(1.-E*E)
      FR=ATAN2(X*DX+Y*DY+Z*DZ,H*(P-R)/P)
      W=ANGLE(RD*(ATAN2((-X*SO+Y*CO)*CXIR+Z*SXIR,X*CO+Y*SO)-FR))
      XI=RD*XIR
      O=ANGLE(RD*ATAN2(SO,CO))
      F=ANGLE(RD*FR)
      RETURN
      END
```

APPENDIX B

```
      SUBROUTINE WELLS(B,U,J2C,A,E,XI,W,O,DT)
C
C     THIS SUBRCUTINE UPDATES THE ARGUMENT OF PERIAPSIS AND THE LONGITUDE
C     OF THE ASCENDING NODE
C
C     B,U,J20 - PLANET RADIUS, GRAVITICNAL CONSTANT, 2ND ZONAL HARMONIC
C     A,E,XI - SEMI-MAJOR AXIS, ECCENTRICITY, INCLINATION
C     W,O,DT - ARGUMENT OF PERIAPSIS, LONGITUDE CF ASCENDING NODE, TIME
C              INCREMENT
C
      REAL J20,N
      DATA PI,DR,RD/3.14159265358979323846,.017453292519943,
     157.2957795130823/
      N=SQRT(U/A**3)
      C=3.*N*J20*B**2
      F=A**2*(1.0-E**2)**2
      WDOT=C/F*(1.0-(5.0/4.0*SIN(XI*DR)**2))*RD
      ODOT=((-C*COS(XI*DR))/(2.0*F))*RD
      W=WDOT*DT+W
      O=ODOT*DT+O
      RETURN
      END
```

APPENDIX C

INPUTS AND OUTPUTS

The definitions of all input and output parameters are given in this appendix along with a sample input and the corresponding output.

Definition of Input Parameters

The following parameters are available for input into the computer program PANDG:

Program symbol	Mathematical symbol	Units	Definition
AO	a	km	Semimajor axis
EO	e		Eccentricity
XIO	i	deg	Inclination
WO	ω	deg	Argument of periapsis
OO	Ω	deg	Longitude of ascending node
FO	f	deg	Initial true anomaly
HAO	H_a	km	Altitude of apoapsis
HPO	H_p	km	Altitude of periapsis
VINF	V_∞	km/sec	Hyperbolic excess velocity of approach hyperbola in areocentric coordinate system
SDEC		deg	Declination of incoming hyperbolic asymptote in areocentric coordinate system
SRA		deg	Right ascension of incoming hyperbolic asymptote in areocentric coordinate system
BETA		deg	Orientation angle of orbital plane (ref. 1)
DATE			Six-dimensional vector defining initial calendar date. The order is: year, month, day, hour, minute, second
JD		days	Initial Julian date
FSTOP		deg	Central-angle stop condition
OSTOP			Orbit-number (integer) stop condition (orbit number changes at periapsis)
TSTOP		sec	Time stop condition
DELT	Δt	sec	Time increment or step
DELF	Δf	deg	True anomaly increment or step
PHI1	ϕ_1	deg	First Sun angle denoting lighting band for photography
PHI2	ϕ_2	deg	Second Sun angle

APPENDIX C

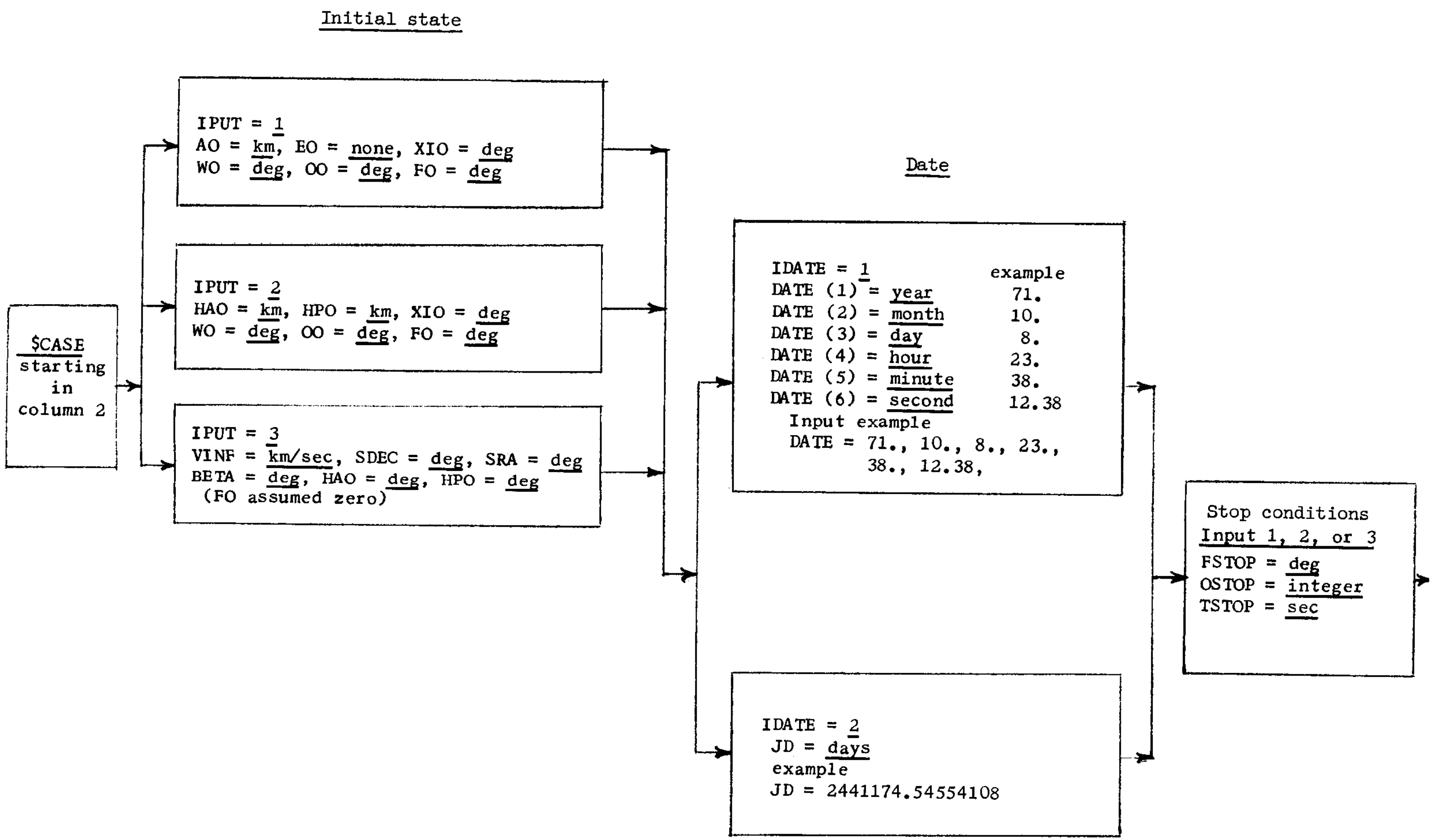
Program symbol	Mathematical symbol	Units	Definition
FPHOTO1	f_1	deg	First true anomaly denoting desirable area for photography
FPHOTO2	f_2	deg	Second true anomaly ($f_2 > f_1$)
LAT1		deg	First latitude denoting desirable area for photography
LAT2		deg	Second latitude
PLAT		deg	Latitude denoting position of a single picture
OVLAP			Ratio of overlap area to area of previous picture
CANGF	ψ_f	deg	Forward camera half-angle (see sketch (b))
CANGS	ψ_s	deg	Side camera half-angle (see sketch (b))
LAMBDA	λ	deg	Out-of-plane angle for nonvertical photography (see sketch (b))
IPUT			Input option for initial state 1 – AO, EO, XIO, WO, OO, FO 2 – HAO, HPO, XIO, WO, OO, FO 3 – VINF, SDEC, SRA, BETA, HAO, HPO
IDATE			Input option for initial time 1 – DATE (calendar date) 2 – JD (Julian date)
IPG			Program mode option 1 – Groundtrack only 2 – Photography only 3 – Photography and groundtrack
ISTEP			Step option 1 – DELT (time increment) 2 – DELF (true anomaly increment)
IFRAME			Option denoting region of photography 1 – Single photograph 2 – Multiple photographs between Sun angles (PHI1, PHI2) 3 – Multiple photographs between true anomalies (FPHOTO1, FPHOTO2) 4 – Multiple photographs between two latitudes (LAT1, LAT2)
ISINGLE			Option denoting position of single photograph 1 – PLAT (photograph taken on latitude) 2 – Not programed

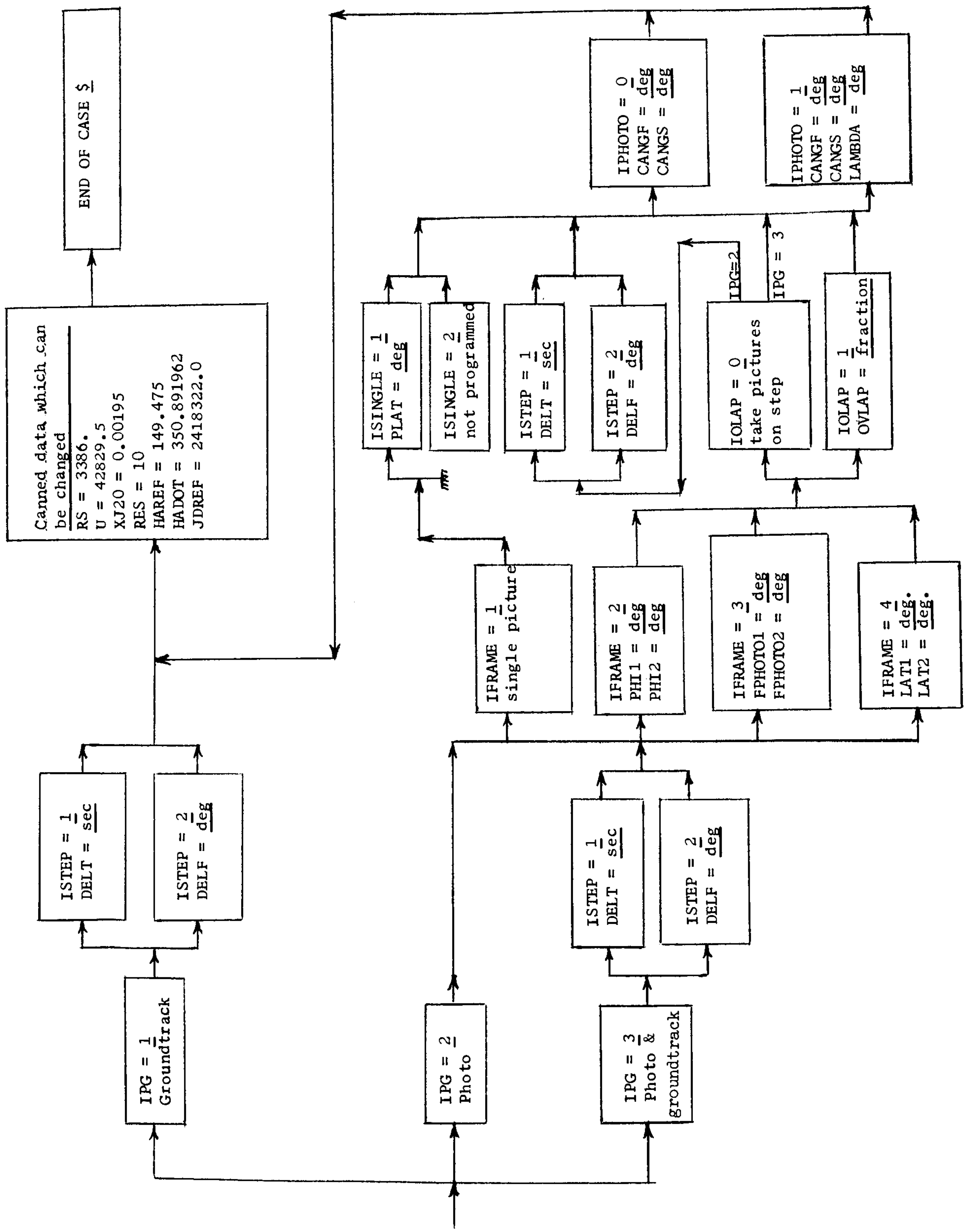
Program symbol	Mathematical symbol	Units	Definition
IOLAP			Multiple photograph option 0 – Successive photographs taken on step (see ISTEP option) 1 – Successive photographs taken on overlap consideration (OVLAP)
IPHOTO			Option denoting type of photography 0 – Vertical photography 1 – Nonvertical photography
RS	r_s	km	Radius of planet
U	μ	km^3/sec^2	Gravitational constant
XJ20	J_{20}	None	Second zonal harmonic of the planet
RES		m/km	Photography resolution constant
HAREF	θ_o	deg	Hour angle at a reference Julian date
JDREF		day	Reference Julian data
HADOT	$\dot{\theta}$	deg/day	Time rate of change of hour angle

Flow Diagram of Input Parameters

A flow diagram of the input parameters has been found to be very helpful. Not all the parameters are defined for a specific case. Only those parameters necessary for a given option must be input. In addition, some numerical values have been stored in the program and are seldom changed. However, an option is available to change these constants when desired. Starting with the input of $CASE, the user can follow the flow diagram and choose the options or sets of input desired. An underline denotes that a numerical value must be input. For most of the parameters, the input units are given. The options are selected by inputing the number which is underlined. The proper input parameters are chosen by proceeding through the flow diagram until the end of the case is reached, at which point a $ is input to terminate the case.

APPENDIX C


Initial state
$CASE starting in column 2
IPUT = 1
AO = km, EO = none, XIO = deg
WO = deg, OO = deg, FO = deg
IPUT = 2
HAO = km, HPO = km, XIO = deg
WO = deg, OO = deg, FO = deg
IPUT = 3
VINF = km/sec, SDEC = deg, SRA = deg
BETA = deg, HAO = deg, HPO = deg
(FO assumed zero)
Date
IDATE = 1 example
DATE (1) = year 71.
DATE (2) = month 10.
DATE (3) = day 8.
DATE (4) = hour 23.
DATE (5) = minute 38.
DATE (6) = second 12.38
Input example
DATE = 71., 10., 8., 23., 38., 12.38,
IDATE = 2
JD = days
example
JD = 2441174.54554108
Stop conditions
Input 1, 2, or 3
FSTOP = deg
OSTOP = integer
TSTOP = sec


IPG = 1
Groundtrack
IPG = 2
Photo
IPG = 3
Photo &
groundtrack
ISTEP = 1
DELT = sec
ISTEP = 2
DELF = deg
Canned data which can be changed
RS = 3386.
U = 42829.5
XJ20 = 0.00195
RES = 10
HAREF = 149.475
HADOT = 350.891962
JDREF = 2418322.0
END OF CASE $
IFRAME = 1
single picture
IFRAME = 2
PHI1 = deg
PHI2 = deg
IFRAME = 3
FPHOTO1 = deg
FPHOTO2 = deg
IFRAME = 4
LAT1 = deg.
LAT2 = deg.
ISINGLE = 1
PLAT = deg
ISINGLE = 2
not programmed
IOLAP = 0
take pictures
on step
IOLAP = 1
OVLAP = fraction
IPG=2
IPG = 3
IPHOTO = 0
CANGF = deg
CANGS = deg
IPHOTO = 1
CANGF = deg
CANGS = deg
LAMBDA = deg

APPENDIX C

Sample Input

The program input is loaded by using a FORTRAN IV namelist. A sample set of input data is as follows:

```
$CASE  IPUT = 2,
       HAO = 30 000.,
       HPO = 1000.,
       XIO = 60.,
       WO = 160.,
       OO = 230.,
       FO = 300.,
       IDATE = 1,
       DATE (1) = 74.,
       DATE (2) = 2.,
       DATE (3) = 16.,
       DATA (4) = 0.,
       DATE (5) = 0.,
       DATE (6) = 0.,
       FSTOP = 400.,
       IPG = 3,
       ISTEP = 2, DELF = 10.,
       IFRAME = 4, LAT1 = 30., LAT2 = -25,
       IOLAP = 1, OVLAP = .25,
       IPHOTO = 0, CANGF = 10, CANGS = 20.$
```

APPENDIX C

Definition of Output Parameters

Definitions of the program output follow:

Program symbol	Mathematical symbol	Units	Definition
JULIAN DATE	JD	day	Current Julian date
CALENDAR DATE			Current calendar date (year, month, day, hour, minute, second)
ORBIT TIME			Time from initial state (day, hour, minute, second)
ORBIT NO			Orbit number (The orbit number is 1 initially and increases by 1 at each periapsis passage.)
LAT	δ	deg	Latitude of the subsatellite point
LONG	α	deg	Longitude of the subsatellite point
ALT	H	km	Altitude of spacecraft
VELI	V	km/sec	Inertial velocity of spacecraft
FPAI	γ	deg	Inertial flight-path angle
PHI	ϕ	deg	Sun angle (angle at subsatellite point between Sun vector and local vertical)
VOHR	V/H	sec^{-1}	V over H ratio or horizontal velocity relative to surface divided by altitude of the spacecraft
HA	θ	deg	Hour angle of Martian vernal equinox as measured from the prime meridian (see sketch (a))
W	ω	deg	Argument of periapsis
O	Ω	deg	Longitude of ascending node
T.A.	f	deg	True anomaly of spacecraft
PHOTO OVERLAP			Ratio of overlap area to area of previous footprint
FOOTPRINT NUMBER			Number of footprint within present sequence of footprints
CENTER			Camera axis
POINT			Points 1, 3, 5, 7 are footprint corner points and 2, 4, 6, 8 are at the midpoint of the sides
LATITUDE	δ	deg	Latitude of respective point
LONGITUDE	α	deg	Longitude of respective point
SUN ANGLE	ϕ	deg	Sun angle of respective point
SLANT RANGE	d	km	Distance between spacecraft and respective point (see sketch (e))
STATIC RESOLUTION		m	Static resolution is slant range multiplied by a constant (RES)

APPENDIX C

Sample Output

The output which resulted from the sample input described previously is presented subsequently. These output data are also presented graphically in figure 1. Because of the lack of space, only the first part of the computer printout follows.

```
$CASE

AO        =  1.88860000E+04    BETA     =  7.86585318E+29    CANGF    =  1.00000000E+01    CANGS    =  2.00000000E+01
DATE(1)   =  7.40000000E+01    DATE(2)  =  2.00000000E+00    DATE(3)  =  1.60000000E+01    DATE(4)  =  0.
DATE(5)   =  0.                DATE(6)  =  0.                DELF     =  1.00000000E+01    DELT     =  0.
EO        =  7.67764482E-01    FO       =  3.00000000E+02    FPHOTO1  =  4.61862172-278    FPHOTO2  =  2.41501531-278
FSTOP     =  4.00000000E+02    HAO      =  3.00000000E+04    HPO      =  1.00000000E+03    IB       =               0
IDATE     =               1    IFRAME   =               4    IMANU    =               0    IOCC     =               0
IOLAP     =               1    IPG      =               3    IPHOTO   =               0    IPUT     =               2
ISINGLE   =               R    ISTEP    =               2                                  JD       =               R
LAMBDA    =  0.                LAT1     =  3.00000000E+01    LAT2     = -2.50000000E+01    OO       =  2.30000000E+02
OSTOP     =           10000    CVLAP    =  2.50000000E-01    PHI1     =  2.24418498-239    PHI2     = -9.20669518+262
PLAT      =  2.06298160-106    RES      =  1.00000000E+01    RS       =  3.38600000E+03    SDEC     = -2.20614006E+72
SRA       = -3.63393157+134    TSTOP    =  1.00000000E+10    U        =  4.28295000E+04    VINF     = -1.10362004E+73
WO        =  1.60000000E+02    XIO      =  6.00000000E+01    XJ20     =  1.95000000E-03

GROUND TRACK AND PHOTO OPTION
INPUT    HPO =  1.00000000E+03  HAO =  3.00000000E+04
CALENDAR  DATE  INPUT
TRUE  ANOMALY  STEP  OF    1.00000000E+01  EMPLOYED
NO  OCCULATION  DATA
VERTICAL  PHOTOGRAPHY
NO  MANEUVERS  MADE
BMATRIX  NOT  COMPUTED
PICTURES  TAKEN  ON     2.50000000E-01  OVERLAP
PICTURES  TAKEN  BETWEEN  LAT1  OF    30.00  AND  LAT2  OF   -25.00
PROGRAM  WILL  STOP  ON  FSTOP =    4.00000000E+02  DEGREES

JULIAN DATE  2442094.50000000  CALENDAR DATE  74  2 16  0  0   0.00  ORBIT TIME  0  0  0  0.00  (SEC)0.          ORBIT NO 1

LAT  =  5.85250511E+01  LONG =  1.70783755E+02  ALT  =  2.21665519E+03  VELI =  3.60849136E+00  FPAI = -2.56625399E+01
PHI  =  5.99803432E+01  VOHR =  1.41328046E-03  HA   =  1.68641645E+02  W    =  1.60000000E+02  O    =  2.30000000E+02
T.A. =  3.00000000E+02

JULIAN DATE  2442094.50321922  CALENDAR DATE  74  2 16  0  4  38.14  ORBIT TIME  0  0  4 38.14  (SEC) 278.141     ORBIT NO 1

LAT  =  5.44686073E+01  LONG =  1.86280923E+02  ALT  =  1.80540659E+03  VELI =  3.77258021E+00  FPAI = -2.14944432E+01
PHI  =  5.31876209E+01  VOHR =  1.87821998E-03  HA   =  1.69771242E+02  W    =  1.60000089E+02  O    =  2.29999646E+02
T.A. =  3.10000000E+02

JULIAN DATE  2442094.50602373  CALENDAR DATE  74  2 16  0  8  40.45  ORBIT TIME  0  0  8 40.45  (SEC) 520.452     ORBIT NO 1

LAT  =  4.85902698E+01  LONG =  1.98350806E+02  ALT  =  1.49606750E+03  VELI =  3.90868895E+00  FPAI = -1.72624501E+01
PHI  =  4.71645187E+01  VOHR =  2.41574326E-03  HA   =  1.70755325E+02  W    =  1.60000166E+02  O    =  2.29999337E+02
T.A. =  3.15999999E+02
```

```
JULIAN DATE  2442094.50853910  CALENDAR DATE  74  2 16  0 12  17.78  ORBIT TIME  0  0 12 17.78  (SEC) 737.780   ORBIT NO 1

LAT  =  4.15605885E+01  LONG =  2.07571591E+02  ALT  =  1.27097552E+03  VELI =  4.01570751E+00  FPAI = -1.29839563E+01
PHI  =  4.22736346E+01  VOHR =  2.98619975E-03  HA   =  1.71637945E+02  W    =  1.60000235E+02  O    =  2.29999060E+02
T.A. =  3.29999999E+02


JULIAN DATE  2442094.51085937  CALENDAR DATE  74  2 16  0 15  38.25  ORBIT TIME  0  0 15 38.25  (SEC) 938.252   ORBIT NO 1

LAT  =  3.38256075E+01  LONG =  2.14786433E+02  ALT  =  1.11796945E+03  VELI =  4.09276998E+00  FPAI = -8.67300963E+00
PHI  =  3.89671445E+01  VOHR =  3.51458015E-03  HA   =  1.72452111E+02  W    =  1.60000299E+02  O    =  2.29998804E+02
T.A. =  3.39999999E+02


JULIAN DATE  2442094.51191200  CALENDAR DATE  74  2 16  0 17   9.20  ORBIT TIME  0  0 17  9.20  (SEC) 1029.20   ORBIT NO 1

LAT  =  2.99997340E+01  LONG =  2.17706214E+02  ALT  =  1.06824825E+03  VELI =  4.11862398E+00  FPAI = -6.62366711E+00
PHI  =  3.80848842E+01  VOHR =  3.72081511E-03  HA   =  1.72821471E+02  W    =  1.60000328E+02  O    =  2.29998689E+02
T.A. =  3.44735608E+02
PHOTO OVERLAP =   0.000          FOOTPRINT NUMBER =   1
```

	LATITUDE	LONGITUDE	SUN ANGLE	SLANT RANGE	STATIC RESOLUTION
CENTER	30.00	217.71	38.08	1068.248	10682.48
POINT 1	36.58	222.34	45.41	1186.760	11867.60
POINT 2	32.60	215.51	38.63	1090.105	10901.05
POINT 3	28.4	209.15	31.94	1186.760	11867.60
POINT 4	25.9[illegible]	211.59	31.42	1161.700	11617.00
POINT 5	23.28	213.66	30.89	1186.760	11867.60
POINT 6	27.37	219.79	37.76	1090.105	10901.05
POINT 7	31.00	226.49	44.63	1186.760	11867.60
POINT 8	33.73	224.32	44.77	1161.700	11617.00

```
JULIAN DATE  2442094.51299261  CALENDAR DATE  74  2 16  0 18  42.56  ORBIT TIME  0  0 18 42.57  (SEC) 1122.57   ORBIT NO 1

LAT  =  2.59137171E+01  LONG =  2.20506857E+02  ALT  =  1.03095446E+03  VELI =  4.13828901E+00  FPAI = -4.47456309E+00
PHI  =  3.76852631E+01  VOHR =  3.88929858E-03  HA   =  1.73200646E+02  W    =  1.60000358E+02  O    =  2.29998570E+02
T.A. =  3.49693252E+02
PHOTO OVERLAP =    .250          FOOTPRINT NUMBER =   2
```

	LATITUDE	LONGITUDE	SUN ANGLE	SLANT RANGE	STATIC RESOLUTION
CENTER	25.91	220.51	37.69	1030.954	10309.54
POINT 1	32.17	224.94	44.49	1144.166	11441.66
POINT 2	28.47	218.55	37.87	1051.865	10518.65
POINT 3	24.64	212.49	31.28	1144.166	11441.66
POINT 4	22.19	214.68	31.19	1120.263	11202.63
POINT 5	19.54	216.53	31.09	1144.166	11441.66
POINT 6	23.33	222.38	37.72	1051.865	10518.65
POINT 7	26.74	228.66	44.35	1144.166	11441.66
POINT 8	29.39	226.70	44.18	1120.263	11202.63

```
JULIAN DATE  2442094.51305889  CALENDAR DATE  74  2 16  0 18  48.29  ORBIT TIME  0  0 18 48.30  (SEC) 1128.30    ORBIT NO 1

LAT  =  2.56586094E+01  LONG =  2.20672765E+02  ALT  =  1.02913195E+03  VELI =  4.13925612E+00  FPAI = -4.34149548E+00
PHI  =  3.76786459E+01  VOHR =  3.89786370E-03  HA   =  1.73223903E+02  W    =  1.60000359E+02  O    =  2.29998562E+02
T.A. =  3.49999997E+02

JULIAN DATE  2442094.51403156  CALENDAR DATE  74  2 16  0 20  12.33  ORBIT TIME  0  0 20 12.34  (SEC) 1212.34    ORBIT NO 1

LAT  =  2.18697362E+01  LONG =  2.23033764E+02  ALT  =  1.00871000E+03  VELI =  4.15013224E+00  FPAI = -2.37793187E+00
PHI  =  3.78277033E+01  VOHR =  3.99611862E-03  HA   =  1.73555206E+02  W    =  1.60000386E+02  O    =  2.29998455E+02
T.A. =  3.54524233E+02
PHOTO OVERLAP =     .250          FOOTPRINT NUMBER =    3
```

	LATITUDE	LONGITUDE	SUN ANGLE	SLANT RANGE	STATIC RESOLUTION
CENTER	21.87	223.03	37.83	1008.710	10087.10
POINT 1	27.92	227.33	44.16	1118.802	11188.02
POINT 2	24.41	221.24	37.68	1029.062	10290.62
POINT 3	20.81	215.40	31.19	1118.802	11188.02
POINT 4	18.35	217.40	31.50	1095.579	10955.79
POINT 5	15.72	219.09	31.80	1118.802	11188.02
POINT 6	19.31	224.76	38.18	1029.062	10290.62
POINT 7	22.58	230.77	44.62	1118.802	11188.02
POINT 8	25.18	228.95	44.16	1095.579	10955.79

```
JULIAN DATE  2442094.51505131  CALENDAR DATE  74  2 16  0 21  40.43  ORBIT TIME  0  0 21 40.45  (SEC) 1300.45    ORBIT NO 1

LAT  =  1.78257653E+01  LONG =  2.25376091E+02  ALT  =  1.00014249E+03  VELI =  4.15471664E+00  FPAI = -3.04360046E-01
PHI  =  3.84787020E+01  VOHR =  4.03876127E-03  HA   =  1.73923027E+02  W    =  1.60000414E+02  O    =  2.29998343E+02
T.A. =  3.59299215E+02
PHOTO OVERLAP =     .250          FOOTPRINT NUMBER =    4
```

	LATITUDE	LONGITUDE	SUN ANGLE	SLANT RANGE	STATIC RESOLUTION
CENTER	17.83	225.38	38.48	1000.142	10001.42
POINT 1	23.77	229.59	44.39	1109.042	11090.42
POINT 2	20.37	223.70	38.01	1020.281	10202.81
POINT 3	16.92	217.97	31.62	1109.042	11090.42
POINT 4	14.44	219.84	32.31	1086.078	10860.78
POINT 5	11.80	221.44	32.97	1109.042	11090.42
POINT 6	15.27	227.01	39.14	1020.281	10202.81
POINT 7	18.45	232.85	45.41	1109.042	11090.42
POINT 8	21.05	231.12	44.68	1086.078	10860.78

```
JULIAN DATE  2442094.51520069  CALENDAR DATE  74  2 16  0 21  53.34  ORBIT TIME  0  0 21 53.36  (SEC) 1313.36    ORBIT NO 1

LAT  =  1.72290434E+01  LONG =  2.25709004E+02  ALT  =  1.00000000E+03  VELI =  4.15479299E+00  FPAI = -1.70031076E-06
PHI  =  3.86156080E+01  VOHR =  4.03951421E-03  HA   =  1.73975445E+02  W    =  1.60000418E+02  O    =  2.29998326E+02
T.A. =  3.59999996E+02
```

JULIAN DATE 2442094.51607256 CALENDAR DATE 74 2 16 0 23 8.67 ORBIT TIME 0 0 23 8.69 (SEC) 1388.69 ORBIT NO 2

LAT = 1.37366225E+01 LONG = 2.27603352E+02 ALT = 1.00484614E+03 VELI = 4.15219816E+00 FPAI = 1.77430382E+00
PHI = 3.96145999E+01 VOHR = 4.01569706E-03 HA = 1.74281376E+02 W = 1.60000442E+02 0 = 2.29998230E+02
T.A. = 4.08556138E+00
PHOTO OVERLAP = .250 FOOTPRINT NUMBER = 5

	LATITUDE	LONGITUDE	SUN ANGLE	SLANT RANGE	STATIC RESOLUTION
CENTER	13.74	227.60	39.61	1004.846	10048.46
POINT 1	19.67	231.80	45.14	1114.400	11144.00
POINT 2	16.31	225.99	38.84	1025.102	10251.02
POINT 3	12.95	220.28	32.55	1114.400	11144.00
POINT 4	10.42	222.09	33.59	1091.294	10912.94
POINT 5	7.74	223.62	34.58	1114.400	11144.00
POINT 6	11.15	229.18	40.56	1025.102	10251.02
POINT 7	14.31	234.96	46.70	1114.400	11144.00
POINT 8	16.93	233.27	45.71	1091.294	10912.94

JULIAN DATE 2442094.51711649 CALENDAR DATE 74 2 16 0 24 38.86 ORBIT TIME 0 0 24 38.89 (SEC) 1478.89 ORBIT NO 2

LAT = 9.55212534E+00 LONG = 2.29775047E+02 ALT = 1.02332894E+03 VELI = 4.14233928E+00 FPAI = 3.88696712E+00
PHI = 4.12235532E+01 VOHR = 3.92639080E-03 HA = 1.74647683E+02 W = 1.60000471E+02 0 = 2.29998115E+02
T.A. = 8.55238831E+00
PHOTO OVERLAP = .250 FOOTPRINT NUMBER = 6

	LATITUDE	LONGITUDE	SUN ANGLE	SLANT RANGE	STATIC RESOLUTION
CENTER	9.55	229.78	41.22	1023.329	10233.29
POINT 1	15.57	234.01	46.41	1135.467	11354.67
POINT 2	12.19	228.18	40.16	1044.047	10440.47
POINT 3	8.85	222.41	33.94	1135.467	11354.67
POINT 4	6.24	224.19	35.31	1111.798	11117.98
POINT 5	3.48	225.69	36.62	1135.467	11354.67
POINT 6	6.90	231.34	42.45	1044.047	10440.47
POINT 7	10.10	237.16	48.51	1135.467	11354.67
POINT 8	12.77	235.47	47.25	1111.798	11117.98

JULIAN DATE 2442094.51734252 CALENDAR DATE 74 2 16 0 24 58.39 ORBIT TIME 0 0 24 58.42 (SEC) 1498.42 ORBIT NO 2

LAT = 8.64875706E+00 LONG = 2.30232967E+02 ALT = 1.02913191E+03 VELI = 4.13925614E+00 FPAI = 4.34149242E+00
PHI = 4.16248198E+01 VOHR = 3.89901405E-03 HA = 1.74726997E+02 W = 1.60000477E+02 0 = 2.29998091E+02
T.A. = 9.59999565E+00

JULIAN DATE 2442094.51820685 CALENDAR DATE 74 2 16 0 26 13.07 ORBIT TIME 0 0 26 13.10 (SEC) 1573.10 ORBIT NO 2

LAT = 5.21404842E+00 LONG = 2.31947645E+02 ALT = 1.05712569E+03 VELI = 4.12446415E+00 FPAI = 6.06549238E+00
PHI = 4.33090677E+01 VOHR = 3.77129528E-03 HA = 1.75030282E+02 W = 1.60000501E+02 0 = 2.29997995E+02
T.A. = 1.39760517E+01
PHOTO OVERLAP = .250 FOOTPRINT NUMBER = 7

	LATITUDE	LONGITUDE	SUN ANGLE	SLANT RANGE	STATIC RESOLUTION
CENTER	5.21	231.95	43.31	1057.126	10571.26
POINT 1	11.43	236.29	48.20	1174.047	11740.47
POINT 2	7.95	230.34	41.95	1078.699	10786.99
POINT 3	4.56	224.40	35.76	1174.047	11740.47
POINT 4	1.85	226.18	37.47	1149.335	11493.35
POINT 5	-1.03	227.69	39.09	1174.047	11740.47
POINT 6	2.47	233.54	44.82	1078.699	10786.99
POINT 7	5.78	239.51	50.86	1174.047	11740.47
POINT 8	8.53	237.77	49.32	1149.335	11493.35

JULIAN DATE 2442094.51937310 CALENDAR DATE 74 2 16 0 27 53.84 ORBIT TIME 0 0 27 53.87 (SEC) 1673.87 ORBIT NO 2

LAT = 6.51593967E-01 LONG = 2.34182139E+02 ALT = 1.10912167E+03 VELI = 4.09734066E+00 FPAI = 8.34745226E+00
PHI = 4.58951881E+01 VOHR = 3.55183370E-03 HA = 1.75439511E+02 W = 1.60000533E+02 0 = 2.29997867E+02
T.A. = 1.92470654E+01
PHOTO OVERLAP = .250 FOOTPRINT NUMBER = 8

	LATITUDE	LONGITUDE	SUN ANGLE	SLANT RANGE	STATIC RESOLUTION
CENTER	.65	234.18	45.90	1109.122	11091.22
POINT 1	7.18	238.70	50.54	1233.546	12335.46
POINT 2	3.53	232.52	44.22	1132.031	11320.31
POINT 3	.03	226.27	38.01	1233.546	12335.46
POINT 4	-2.85	228.11	40.07	1207.192	12071.92
POINT 5	-5.88	229.67	42.01	1233.546	12335.46
POINT 6	-2.23	235.85	47.71	1132.031	11320.31
POINT 7	1.27	242.09	53.79	1233.546	12335.46
POINT 8	4.14	240.26	51.96	1207.192	12071.92

JULIAN DATE 2442094.51954202 CALENDAR DATE 74 2 16 0 28 8.43 ORBIT TIME 0 0 28 8.47 (SEC) 1688.47 ORBIT NO 2

LAT = -4.62012451E-04 LONG = 2.34499331E+02 ALT = 1.11796938E+03 VELI = 4.09277001E+00 FPAI = 8.67300708E+00
PHI = 4.62939359E+01 VOHR = 3.51661162E-03 HA = 1.75498784E+02 W = 1.60000538E+02 0 = 2.29997848E+02
T.A. = 1.99999956E+01

JULIAN DATE 2442094.52065194 CALENDAR DATE 74 2 16 0 29 44.33 ORBIT TIME 0 0 29 44.36 (SEC) 1784.36 ORBIT NO 2

LAT = -4.21402820E+00 LONG = 2.36547585E+02 ALT = 1.18401769E+03 VELI = 4.05905163E+00 FPAI = 1.07743429E+01
PHI = 4.90265177E+01 VOHR = 3.27109344E-03 HA = 1.75838243E+02 W = 1.60000568E+02 0 = 2.29997726E+02
T.A. = 2.48668389E+01
PHOTO OVERLAP = .250 FOOTPRINT NUMBER = 9

	LATITUDE	LONGITUDE	SUN ANGLE	SLANT RANGE	STATIC RESOLUTION
CENTER	-4.21	236.55	49.03	1184.018	11840.18
POINT 1	2.80	241.34	53.47	1319.563	13195.63
POINT 2	-1.14	234.77	47.00	1208.897	12088.97
POINT 3	-4.84	228.05	40.70	1319.563	13195.63
POINT 4	-7.93	230.01	43.13	1290.763	12907.63
POINT 5	-11.20	231.67	45.43	1319.563	13195.63
POINT 6	-7.29	238.34	51.17	1208.897	12088.97
POINT 7	-3.49	245.03	57.39	1319.563	13195.63
POINT 8	-.44	243.03	55.22	1290.763	12907.63

JULIAN DATE 2442094.52186225 CALENDAR DATE 74 2 16 0 31 28.90 ORBIT TIME 0 0 31 28.94 (SEC) 1888.94 ORBIT NO 2

LAT = -8.64968037E+00 LONG = 2.38723333E+02 ALT = 1.27097541E+03 VELI = 4.01570756E+00 FPAI = 1.29839539E+01
PHI = 5.21632419E+01 VOHR = 2.98869177E-03 HA = 1.76312934E+02 W = 1.60000602E+02 0 = 2.29997593E+02
T.A. = 2.99999955E+01

JULIAN DATE 2442094.52210042 CALENDAR DATE 74 2 16 0 31 49.48 ORBIT TIME 0 0 31 49.52 (SEC) 1909.52 ORBIT NO 2

LAT = -9.50057402E+00 LONG = 2.39145702E+02 ALT = 1.28981841E+03 VELI = 4.00646762E+00 FPAI = 1.34077773E+01
PHI = 5.27921293E+01 VOHR = 2.93279461E-03 HA = 1.76396504E+02 W = 1.60000608E+02 O = 2.29997567E+02
T.A. = 3.05867245E+01
PHOTO OVERLAP = .250 FOOTPRINT NUMBER = 10

	LATITUDE	LONGITUDE	SUN ANGLE	SLANT RANGE	STATIC RESOLUTION
CENTER	-9.50	239.15	52.79	1289.818	12898.18
POINT 1	-1.78	244.33	57.09	1441.714	14417.14
POINT 2	-6.16	237.17	50.37	1317.574	13175.74
POINT 3	-10.17	229.75	43.84	1441.714	14417.14
POINT 4	-13.57	231.89	46.72	1409.292	14092.92
POINT 5	-17.14	233.72	49.44	1441.714	14417.14
POINT 6	-12.83	241.16	55.33	1317.574	13175.74
POINT 7	-8.58	248.50	61.82	1441.714	14417.14
POINT 8	-5.29	246.22	59.23	1409.292	14092.92

JULIAN DATE 2442094.52381086 CALENDAR DATE 74 2 16 0 34 17.26 ORBIT TIME 0 0 34 17.30 (SEC) 2057.30 ORBIT NO 2

LAT = -1.53778806E+01 LONG = 2.42137304E+02 ALT = 1.44056352E+03 VELI = 3.93441466E+00 FPAI = 1.63375259E+01
PHI = 5.73473730E+01 VOHR = 2.54129415E-03 HA = 1.76996686E+02 W = 1.60000655E+02 O = 2.29997378E+02
T.A. = 3.78301991E+01
PHOTO OVERLAP = .250 FOOTPRINT NUMBER = 11

	LATITUDE	LONGITUDE	SUN ANGLE	SLANT RANGE	STATIC RESOLUTION
CENTER	-15.38	242.14	57.35	1440.564	14405.64
POINT 1	-6.60	247.87	61.58	1617.085	16170.85
POINT 2	-11.66	239.85	54.43	1472.605	14726.05
POINT 3	-16.15	231.33	47.50	1617.085	16170.85
POINT 4	-19.97	233.79	50.94	1579.151	15791.51
POINT 5	-24.00	235.90	54.18	1617.085	16170.85
POINT 6	-19.07	244.51	60.38	1472.605	14726.05
POINT 7	-14.09	252.64	67.31	1617.085	16170.85
POINT 8	-10.49	250.11	64.20	1579.151	15791.51

JULIAN DATE 2442094.52437754 CALENDAR DATE 74 2 16 0 35 6.22 ORBIT TIME 0 0 35 6.27 (SEC) 2106.27 ORBIT NO 2

LAT = -1.72299644E+01 LONG = 2.43116258E+02 ALT = 1.49606736E+03 VELI = 3.90868901E+00 FPAI = 1.72624479E+01
PHI = 5.88533143E+01 VOHR = 2.41827118E-03 HA = 1.77195528E+02 W = 1.60000671E+02 O = 2.29997316E+02
T.A. = 3.99999954E+01

JULIAN DATE 2442094.52595095 CALENDAR DATE 74 2 16 0 37 22.16 ORBIT TIME 0 0 37 22.21 (SEC) 2242.21 ORBIT NO 2

LAT = -2.21114898E+01 LONG = 2.45815942E+02 ALT = 1.66302524E+03 VELI = 3.83375299E+00 FPAI = 1.97075108E+01
PHI = 6.29692787E+01 VOHR = 2.10111682E-03 HA = 1.77747626E+02 W = 1.60000714E+02 O = 2.29997143E+02
T.A. = 4.57617314E+01
PHOTO OVERLAP = .250 FOOTPRINT NUMBER = 12

	LATITUDE	LONGITUDE	SUN ANGLE	SLANT RANGE	STATIC RESOLUTION
CENTER	-22.11	245.82	62.97	1663.025	16630.25
POINT 1	-11.68	252.32	67.23	1878.862	18788.62
POINT 2	-17.87	242.98	59.38	1701.800	17018.00
POINT 3	-23.10	232.71	51.76	1878.862	18788.62
POINT 4	-27.54	235.70	55.97	1831.986	18319.86
POINT 5	-32.24	238.28	59.90	1878.862	18788.62
POINT 6	-26.30	249.83	66.66	1701.800	17018.00
POINT 7	-20.10	258.65	74.36	1878.862	18788.62
POINT 8	-16.10	255.15	70.52	1831.986	18319.86

```
JULIAN DATE  2442094.52718200  CALENDAR DATE  74  2 16  0 39   8.52  ORBIT TIME  0  0 39  8.56  (SEC) 2348.58    ORBIT NO 2

LAT  = -2.56595249E+01  LONG =  2.47919989E+02  ALT  =  1.80540641E+03  VELI =  3.77258029E+00  FPAI =  2.14944411E+01
PHI  =  6.60880783E+01  VOHR =  1.88045245E-03  HA   =  1.78179590E+02  W    =  1.60000748E+02  O    =  2.29997007E+02
T.A. =  4.99999953E+01

JULIAN DATE  2442094.52888286  CALENDAR DATE  74  2 16  0 41  35.48  ORBIT TIME  0  0 41 35.53  (SEC) 2495.53    ORBIT NO 2

LAT  = -3.01712295E+01  LONG =  2.50831743E+02  ALT  =  2.01647310E+03  VELI =  3.68615318E+00  FPAI =  2.37850161E+01
PHI  =  7.02062352E+01  VOHR =  1.61542834E-03  HA   =  1.78776409E+02  W    =  1.60000795E+02  O    =  2.29996820E+02
T.A. =  5.54733980E+01
PHOTO OVERLAP =    .250          FOOTPRINT NUMBER =  13
           LATITUDE          LONGITUDE         SUN ANGLE        SLANT RANGE       STATIC RESOLUTION
CENTER      -30.17            250.83             70.21           2016.473             20164.73
POINT 1     -16.92            258.37             74.66           2302.565             23025.65
POINT 2     -25.15            246.95             65.63           2066.952             20669.52
POINT 3     -31.58            233.49             56.72           2302.565             23025.65
POINT 4     -37.06            237.46             62.09           2239.271             22392.71
POINT 5     -42.86            240.97             67.05           2302.565             23025.65
POINT 6     -35.07            255.13             74.88           2066.952             20669.52
POINT 7     -26.59            267.33             83.93           2302.565             23025.65
POINT 8     -22.11            262.32             78.95           2239.271             22392.71

JULIAN DATE  2442094.53040116  CALENDAR DATE  74  2 16  0 43  46.66  ORBIT TIME  0  0 43 46.72  (SEC) 2626.72    ORBIT NO 2

LAT  = -3.38265094E+01  LONG =  2.53448566E+02  ALT  =  2.21665496E+03  VELI =  3.60849145E+00  FPAI =  2.56625379E+01
PHI  =  7.36731841E+01  VOHR =  1.41501118E-03  HA   =  1.79309166E+02  W    =  1.60000837E+02  O    =  2.29996653E+02
T.A. =  5.99999952E+01

JULIAN DATE  2442094.53368567  CALENDAR DATE  74  2 16  0 48  30.44  ORBIT TIME  0  0 48 30.50  (SEC) 2910.50    ORBIT NO 2

LAT  = -4.06104878E+01  LONG =  2.59206501E+02  ALT  =  2.67832864E+03  VELI =  3.44343841E+00  FPAI =  2.92312595E+01
PHI  =  8.04699516E+01  VOHR =  1.07835489E-03  HA   =  1.80461676E+02  W    =  1.60000927E+02  O    =  2.29996291E+02
T.A. =  6.87290904E+01
PHOTO OVERLAP =    .250          FOOTPRINT NUMBER =  14
           LATITUDE          LONGITUDE         SUN ANGLE        SLANT RANGE       STATIC RESOLUTION
CENTER      -40.61            259.21             80.47           2678.329             26783.29
POINT 1     -21.46            267.76             85.48           3125.306             31253.06
POINT 2     -34.31            252.74             74.14           2754.114             27541.14
POINT 3     -42.97            231.81             62.32           3125.306             31253.06
POINT 4     -50.58            238.00             69.87           3022.225             30222.25
POINT 5     -58.61            243.80             76.62           3125.306             31253.06
POINT 6     -46.47            266.97             86.87           2754.114             27541.14
POINT 7     -32.55            282.75             98.88           3125.306             31253.06
POINT 8     -27.85            274.27             91.65           3022.225             30222.25

JULIAN DATE  2442094.53420950  CALENDAR DATE  74  2 16  0 49  15.70  ORBIT TIME  0  0 49 15.83  (SEC) 2955.83    ORBIT NO 2

LAT  = -4.15614594E+01  LONG =  2.60141318E+02  ALT  =  2.75487629E+03  VELI =  3.41777639E+00  FPAI =  2.97442667E+01
PHI  =  8.14683216E+01  VOHR =  1.03474996E-03  HA   =  1.80645485E+02  W    =  1.60000942E+02  O    =  2.29996233E+02
T.A. =  6.99999950E+01
```

```
JULIAN DATE  2442094.53886271  CALENDAR DATE  74  2 16  0 55  57.74  ORBIT TIME  0  0 55 57.87  (SEC) 3357.87   ORBIT NO 2

LAT  = -4.65910749E+01  LONG = 2.68612077E+02  ALT  = 3.45532317E+03  VELI = 3.20203569E+00  FPAI = 3.37094213E+01
PHI  =  8.93648682E+01  VOHR = 7.36768458E-04  HA   = 1.82278256E+02  W    = 1.60001070E+02  0    = 2.29995721E+02
T.A. =  7.99999949E+01

JULIAN DATE  2442094.54474087  CALENDAR DATE  74  2 16  1  4  25.61  ORBIT TIME  0  1  4 25.74  (SEC) 3865.74   ORBIT NO 2

LAT  = -5.44692773E+01  LONG = 2.79603644E+02  ALT  = 4.36741448E+03  VELI = 2.96312601E+00  FPAI = 3.75157719E+01
PHI  =  9.72691007E+01  VOHR = 5.10928908E-04  HA   = 1.84340857E+02  W    = 1.60001232E+02  0    = 2.29995074E+02
T.A. =  8.99999948E+01

JULIAN DATE  2442094.55243276  CALENDAR DATE  74  2 16  1 15  30.19  ORBIT TIME  0  1 15 30.32  (SEC) 4530.32   ORBIT NO 2

LAT  = -5.85254652E+01  LONG = 2.93531585E+02  ALT  = 5.56011906E+03  VELI = 2.70318312E+00  FPAI = 4.11018020E+01
PHI  =  1.05088478E+02  VOHR = 3.44830761E-04  HA   = 1.87039879E+02  W    = 1.60001443E+02  0    = 2.29994227E+02
T.A. =  9.99999947E+01

JULIAN DATE  2442094.56287310  CALENDAR DATE  74  2 16  1 30  32.24  ORBIT TIME  0  1 30 32.37  (SEC) 5432.37   ORBIT NO 2

LAT  = -6.00000000E+01  LONG = 3.09293219E+02  ALT  = 7.12840034E+03  VELI = 2.42467235E+00  FPAI = 4.43737413E+01
PHI  =  1.12718206E+02  VOHR = 2.26297045E-04  HA   = 1.90703309E+02  W    = 1.60001731E+02  0    = 2.29993078E+02
T.A. =  1.09999995E+02

JULIAN DATE  2442094.57757664  CALENDAR DATE  74  2 16  1 51  42.62  ORBIT TIME  0  1 51 42.75  (SEC) 6702.75   ORBIT NO 2

LAT  = -5.85244376E+01  LONG = 3.23558102E+02  ALT  = 9.19830566E+03  VELI = 2.13049779E+00  FPAI = 4.71809689E+01
PHI  =  1.20025319E+02  VOHR = 1.44434686E-04  HA   = 1.95862663E+02  W    = 1.60002135E+02  0    = 2.29991459E+02
T.A. =  1.19999994E+02

JULIAN DATE  2442094.59901716  CALENDAR DATE  74  2 16  2 22  35.08  ORBIT TIME  0  2 22 35.22  (SEC) 8555.22   ORBIT NO 2

LAT  = -5.44672660E+01  LONG = 3.32659543E+02  ALT  = 1.19221135E+04  VELI = 1.82424482E+00  FPAI = 4.92659268E+01
PHI  =  1.26827747E+02  VOHR = 8.99810715E-05  HA   = 2.03385970E+02  W    = 1.60002725E+02  0    = 2.29989098E+02
T.A. =  1.29999994E+02

JULIAN DATE  2442094.63115627  CALENDAR DATE  74  2 16  3  8  51.90  ORBIT TIME  0  3  8 52.03  (SEC) 11332.0   ORBIT NO 2

LAT  = -4.85880183E+01  LONG = 3.34432960E+02  ALT  = 1.54394415E+04  VELI = 1.51075518E+00  FPAI = 5.01533177E+01
PHI  =  1.32867861E+02  VOHR = 5.53347016E-05  HA   = 2.14663324E+02  W    = 1.60003610E+02  0    = 2.29985560E+02
T.A. =  1.39999994E+02
```

```
JULIAN DATE  2442094.67985855  CALENDAR DATE  74  2 16  4 18  59.78  ORBIT TIME  0  4 18 59.91  (SEC) 15539.9    ORBIT NO 2

LAT   = -4.15570839E+01  LONG =  3.27442315E+02  ALT   =  1.97518577E+04  VELI =  1.19763250E+00  FPAI =  4.88818157E+01
PHI   =  1.37788843E+02  VOHR =  3.44685401E-05  HA    =  2.31752565E+02  W    =  1.60004951E+02  O    =  2.29980198E+02
T.A. =  1.49999994E+02

JULIAN DATE  2442094.75208735  CALENDAR DATE  74  2 16  6  3    .35  ORBIT TIME  0  6  3   .48  (SEC) 21780.5    ORBIT NO 2

LAT   = -3.38203087E+01  LONG =  3.10119720E+02  ALT   =  2.44501711E+04  VELI =  8.99702264E-01  FPAI =  4.33120496E+01
PHI   =  1.41145420E+02  VOHR =  2.28160339E-05  HA    =  2.57097071E+02  W    =  1.60006939E+02  O    =  2.29972245E+02
T.A. =  1.59999994E+02

JULIAN DATE  2442094.85161319  CALENDAR DATE  74  2 16  8 26  19.38  ORBIT TIME  0  8 26 19.51  (SEC) 30379.5    ORBIT NO 2

LAT   = -2.56508588E+01  LONG =  2.81845237E+02  ALT   =  2.84033715E+04  VELI =  6.53290823E-01  FPAI =  2.86620048E+01
PHI   =  1.42505646E+02  VOHR =  1.71697655E-05  HA    =  2.92019888E+02  W    =  1.60009678E+02  O    =  2.29961288E+02
T.A. =  1.69999994E+02

JULIAN DATE  2442094.86413829  CALENDAR DATE  74  2 16  8 44  21.55  ORBIT TIME  0  8 44 21.68  (SEC) 31461.7    ORBIT NO 2

LAT   = -2.47281601E+01  LONG =  2.78124905E+02  ALT   =  2.87238086E+04  VELI =  6.32375334E-01  FPAI =  2.61764313E+01
PHI   =  1.42520712E+02  VOHR =  1.68190563E-05  HA    =  2.96414844E+02  W    =  1.60010023E+02  O    =  2.29959909E+02
T.A. =  1.71106698E+02

PHOTO FOOTPRINT OFF PLANET
```

REFERENCES

1. Green, Richard N.: Investigation of Occultation and Imagery Problems for Orbital Missions to Venus and Mars. NASA TN D-5104, 1969.

2. Melbourne, William G.; Mulholland, J. Derral; Sjogren, William L.; and Sturms, Francis M., Jr.: Constants and Related Information for Astrodynamic Calculations, 1968. Tech. Rep. No. 32-1306 (Contract NAS-7-100), Jet Propulsion Lab., California Inst. Technol., July 15, 1968.

3. Escobal, Pedro Ramon: Methods of Orbit Determination. John Wiley & Sons, Inc., 1965.

4. Welchons, A. M.; and Krickenberger, W. R.: Solid Geometry. Rev. ed., Ginn and Co., c.1950.

NASA-Langley, 1970 — 8 L-7214